

# **A Holistic Solution for Reliability of 3D Parallel Systems**

by

Javad Bagherzadeh

A dissertation submitted in partial fulfillment  
of the requirements for the degree of  
Doctor of Philosophy  
(Computer Science and Engineering)  
in the University of Michigan  
2021

Doctoral Committee:

Assistant Professor Ronald Dreslinski, Jr., Chair  
Professor Wei Lu  
Professor Trevor N. Mudge  
Professor Thomas F. Wenisch

Javad Bagherzadeh

[javadb@umich.edu](mailto:javadb@umich.edu)

ORCID iD: [0000-0002-4675-0697](https://orcid.org/0000-0002-4675-0697)

© Javad Bagherzadeh 2021

*Dedicated to my parents, Mosayyeb and Kobra, for their boundless love and support.*

## ACKNOWLEDGMENTS

First and foremost, I would like to thank my adviser, Ronald Dreslinski, for taking me as a student and providing me with the support and guidance that I needed to succeed in graduate school. The experience I gained is worthwhile and invaluable, and this research work would not have been possible if not for the continued encouragement and opportunities that Ron has provided.

Thanks are also due to my doctoral committee members, Prof. Trever Mudge, Prof. Wei Lu, and Prof. Tom Wenisch for their insightful feedback and constructive suggestions on the topics of my thesis.

I am also greatly thankful to Aporva Amarnath, Jielun Tan, Subhankar Pal and all other members of the CADRE for being wonderful research collaborators and co-authors on several pieces of my work.

I am grateful to the friends that I made during my time at Michigan: Mohammad Khodabakhsh, Mohammadreza, Ahmad, Hossein, Heydar, Morteza, Ehsan, Ali, Shahab, Rashin, Mehrdad, Donya, Hamed, Amir, Zakaria, Vishishtha, Sugandha, Disha, Ameya, Jinal, Doowon, Navid, Saeed, Reza, and Mehdi.

Finally, I would like to express my sincere gratitude to my parents and my siblings Sajad and Saleh for their continued support and love.

## TABLE OF CONTENTS

<b>DEDICATION</b> . . . . .	ii
<b>ACKNOWLEDGMENTS</b> . . . . .	iii
<b>LIST OF FIGURES</b> . . . . .	viii
<b>LIST OF TABLES</b> . . . . .	xii
<b>ABSTRACT</b> . . . . .	xiii
<b>CHAPTER</b>	
<b>I. Introduction</b> . . . . .	1
1.1 Emerging 3D Technology . . . . .	2
1.1.1 Monolithic 3D . . . . .	2
1.1.2 Carbon Nanotube Field Effect Transistors . . . . .	3
1.2 Reliability Challenge . . . . .	5
1.3 Sustaining a disruptive technology . . . . .	7
1.4 Previous solutions . . . . .	8
1.4.1 Treatment Techniques . . . . .	9
1.4.2 Lifetime Management Techniques . . . . .	10
1.4.3 Frankenstein Solutions . . . . .	10
1.4.4 Shortcomings of previous solutions . . . . .	11
1.5 Contributions of this thesis . . . . .	12
1.5.1 3DFAR: A New fabric for Reliability of Processors in 3D . . . . .	13
1.5.2 R2D3: A Holistic Solution . . . . .	14
1.5.3 3DTUBE: A Design Framework for High-Variation Carbon Nanotube-based Transistor Technology . . . . .	15
<b>II. Background</b> . . . . .	17
2.1 Lifetime Failure Models . . . . .	17
2.2 3D Circuits and Challenges . . . . .	20

2.2.1	CNT Technology . . . . .	22
2.2.2	Systems Using M3D and CNT Technology . . . . .	23
<b>III.</b>	<b>A Three-Dimensional Fabric for Reliable Multi-Core Processors . . . . .</b>	<b>26</b>
3.1	Introduction . . . . .	27
3.2	Architecture . . . . .	29
3.3	Latency of Interconnect Switches . . . . .	30
3.3.1	Flexible Deployment of 3DFAR . . . . .	31
3.3.2	Number of Design Layers . . . . .	32
3.4	Interconnect Switch Design . . . . .	32
3.4.1	Middle-Layer Interconnect . . . . .	34
3.4.2	Vertically Distributed Interconnect . . . . .	34
3.4.3	Vertical Bus Interconnect . . . . .	35
3.5	3DFAR System-Level Operation . . . . .	36
3.5.1	Faulty Registers and Register Files . . . . .	37
3.5.2	Faulty Local Cache Units . . . . .	38
3.6	Experimental Evaluation . . . . .	38
3.6.1	Fault Model . . . . .	39
3.7	Performance in Presence of Faults . . . . .	40
3.7.1	3DFAR Layers Depth . . . . .	41
3.8	Vertical Connections Reliability . . . . .	43
3.9	Related Work . . . . .	46
3.10	Summary . . . . .	48
<b>IV.</b>	<b>R2D3: A Holistic Solution for Reliability of 3D Parallel Systems . . . . .</b>	<b>49</b>
4.1	Introduction . . . . .	50
4.2	Motivation . . . . .	53
4.2.1	Opportunities born from challenges . . . . .	56
4.2.2	Fast 3D reconfiguration fabric . . . . .	56
4.3	Architecture . . . . .	57
4.3.1	Interconnect Switch Design . . . . .	57
4.3.2	Adaptation for Out-of-Order Architectures . . . . .	58
4.3.3	Cluster Size and Number of Design Layers . . . . .	59
4.4	Fault Detection and Diagnosis . . . . .	60
4.5	Fault Detection . . . . .	61
4.6	Fault Diagnosis . . . . .	63
4.7	Repair . . . . .	64
4.8	Assumptions and Limitations . . . . .	64
4.8.1	3D vs 2D . . . . .	64
4.8.2	Addressing Soft Faults . . . . .	65
4.8.3	Caches and System Level Operation . . . . .	65
4.8.4	Requirements of Architecture . . . . .	65
4.8.5	Crossbar Reliability . . . . .	66

<b>V. R2D3 Aging-aware Policy</b>	67
5.1 Intuition on the Proposed Lifetime Management	67
5.2 Random Reconfiguration (R2D3-Lite)	71
5.3 R2D3	72
<b>VI. R2D3 Evaluation and Results</b>	74
6.1 Experimental Methodology	74
6.1.1 Modeling Aging and Mean Time to Failure	77
6.1.2 Fault Model	77
6.2 Physical Design	78
6.2.1 Performance in presence of faults	79
6.3 Fault detection and diagnosis	81
6.3.1 Length of testing period ( $T_{test}$ )	82
6.3.2 Power Overhead and Epoch Size	84
6.3.3 Worst Case Overhead	84
6.4 Lifetime Management	85
6.4.1 Thermal Simulation	85
6.4.2 NBTI Degradation	86
6.5 Related Work	91
6.5.1 Fault Detection	91
6.5.2 NBTI and Lifetime Management Techniques	93
6.5.3 Comparison to the Frankenstein Solutions	95
6.6 Summary	95
<b>VII. Reliability for CNT Technology</b>	97
7.1 Introduction	97
7.2 Motivation	99
7.3 Variation Model	100
7.3.1 CNT Distribution	102
7.3.2 Variation Suite	103
7.4 Architecture	104
7.5 Design Flow for High-variation Technology	106
7.5.1 Standard Cell Library	107
7.5.2 Library Generation	107
7.5.3 Design Methodology	108
7.6 Evaluation	109
7.6.1 Performance and EDP Analysis	109
7.6.2 Frequency and Area Overhead	110
7.7 Summary	111
<b>VIII. Conclusion and Future Directions</b>	113

8.1 Future Directions . . . . .	115
<b>BIBLIOGRAPHY . . . . .</b>	<b>116</b>



## LIST OF FIGURES

### Figure

1.1	Comparison between TSV-based and monolithic 3D integration using MIVs. TSV integration adds overhead and limits the expected benefit of 3D integration. Higher integration density can be achieved through monolithic integration [26]. . . . .	3
1.2	Path to sustenance for a disruptive technology [44] . . . . .	8
1.3	Required features for a holistic reliability solution and existing solutions built in isolation for a particular category . . . . .	9
2.1	Monolithic 3D integration using MIVs [77] . . . . .	21
2.2	Monolithically integrated 3D system for abundant data computing [80]. .	24
3.1	Schematic of the 3DFAR approach. In 3DFAR multi-core architectures, corresponding pipeline stages are stacked vertically, while specialized crossbar units are inserted between each pair of stages. In a four-faults situation as the one illustrated, a regular 2D CMP would be completely disabled. In contrast, 3DFAR can dynamically reconfigure to connect healthy units as shown with the wideband lines, providing the computing power of 3 complete cores. . . . .	29
3.2	Propagation delays through inter-stage interconnects for a 4-core planar layout vs. a 3D stacked layout. The large delay for the planar design is mostly due to the much longer wires required to connect corresponding units. . . . .	32
3.3	Middle-layer and vertically distributed interconnect switches. a) The middle-layer solution entails the worst area overhead, concentrated in the middle layers. Delay overhead does not exceed $2 \times \text{traverse\_delay}(\# \text{layers} / 2)$ . b) The vertically distributed design balances area at the switch granularity, delay overhead can be as much as the cost of traversing all layers. . .	33
3.4	Middle-layer and vertically distributed interconnect switches. a) The middle-layer solution entails the worst area overhead, concentrated in the middle layers. Delay overhead does not exceed $2 \times \text{traverse\_delay}(\# \text{layers} / 2)$ . b) The vertically distributed design balances area at the switch granularity, delay overhead can be as much as the cost of traversing all layers. . .	35

3.5	The vertical bus interconnect a) uses a set of multiplexers in each layer to select which input to route to a pipeline stage, from one of the vertical busses or from the same layer. b) Example illustrating that no more than $\leq \lfloor \#layers/2 \rfloor$ vertical busses are needed. . . . .	37
3.6	Layout of one 3D layer including a complete core (no cache) and all vertical bus switches. . . . .	39
3.7	Performance of 3DFAR against state-of-the-art solutions and a baseline 2D design over varying fault numbers. . . . .	41
3.8	Frequency and area of 3DFAR for a varying number of 3D layers, using the proposed interconnect solutions. As the number of stacked cores increases, the area overhead increases due to larger crossbars and more MIVs and the frequency will decrease as the total frequency will decline due to larger crossbars and more MIVs. . . . .	43
3.9	Average IPC when cluster size is increasing. The total stacked cores kept at 16 and 10000 random fault cases is tested exhaustively for each point of x-axis. . . . .	44
3.10	Average IPC for different total MIV buses. The total stacked cores kept and the cluster size is kept at 16 and 10000 random fault cases is tested exhaustively for each point of x-axis. . . . .	45
3.11	A conceptual figure on how spare MIV methods work (a) Operation when all f-MIVs are fault-free (b) Operation when the third f-MIV is faulty. . .	46
3.12	Average number of working vertical buses for different number of added spare MIVs per each 100s for different number of inserted faults in vertical connection structure for a 4-core structure. . . . .	47
4.1	Required features for holistic reliability solution and existing solutions are built in isolation for a particular category . . . . .	51
4.2	Schematic of the R2D3 Engine, where corresponding pipeline stages are stacked vertically and crossbars are inserted between consecutive stages. In this four-fault situation, our solution dynamically reconfigures to connect healthy units as shown with the red and green stripes, providing the compute power of 2 complete cores. Stages in orange are leftovers and still functional and are swap out working stages to distribute wearout and to detect, diagnose and repair faults. . . . .	54
4.3	Schematic showing high level components of R2D3, where corresponding pipeline stages are stacked vertically and vertical buses that function as crossbars are inserted between consecutive stages. At the end of each epoch, detection circuitry can access the inputs and outputs of all layers to check the output and the R2D3 Reconfiguration Controller manages the components. . . . .	58
4.4	R2D3 fault detection, diagnosis and tolerance mechanism in a flowchart (left) and an example on how pipeline stages in a 4-layer design reconfigure (right) . . . . .	62

5.1	Temperature-based $V_{th}$ degradation of different cores in a 4-core processor in 3D over the duration of 100 seconds with equal on and off time periods (activity factors). All cores are on and then off for periods 10 second and the system does not have any faults. The difference between the hot spot temperature of top and bottom layers is 28 degrees. . . . .	68
5.2	Temperature-based $V_{th}$ degradation of different cores in a 4-core processor in 3D over the duration of 100 seconds with unequal activity factors. The activity factor of each cores is changed so that all cores have the same $V_{th}$ degradation rate. . . . .	69
5.3	$V_{th}$ degradation of different cores in a 4-core processor in 3D over the duration of 100 seconds by changing their activity factors. . . . .	70
6.1	Flow chart of the proposed reliability evaluation methodology for NBTI failure mechanism. . . . .	76
6.2	Layout of one 3D layer including a complete core (no cache) and all vertical bus switches. . . . .	79
6.3	Performance of R2D3 with a varying number of concurrent faults for an in-order 8-core 8-layer design . . . . .	80
6.4	Performance of R2D3 with a varying number of concurrent faults an OOO 8-core 8-layer design . . . . .	81
6.5	Breakdown of different types of faults for each unit. Total illustrates the additive results for all pipeline stages in the stage-level approach and core-level shows the coverage for solutions that have fault detection at a core granularity. . . . .	82
6.6	Breakdown of average percentage of detected faults of detectable permanent faults by length of testing period $T_{test}$ for each unit. 96% of the faults are detected within 5K cycles. . . . .	83
6.7	Worst case performance degradation (left) and power overhead (right) of reconfiguration policy for varying epoch sizes. For the worst case scenario, we assume all 8 cores are always busy and we have to halt one of them to test and run instructions in parallel. For our epoch size (5M cycles) the performance degradation is only 1%. . . . .	83
6.8	Average temperature map of the hottest layer (farthest from the heat-sink) when running different workloads in a loop for (a) Static 3D, (b) R2D3-Lite and (c) R2D3. . . . .	86
6.9	Average $V_{th}$ degradation map of all layers after 8 years when running different workloads in a loop for (a) Static 3D, (b) R2D3-Lite and (c) R2D3. . . . .	86
6.10	$V_{th}$ degradation for NoRecon, Static, R2D3-Lite and R2D3 over period of 8 years. Degradation for the NoRecon and Static is same, although, Static has much better performance in the presence of faults and failures over the same period of time. . . . .	87
6.11	Mean Time To Failure for NoRecon 3D, Static, R2D3-Lite and R2D3 over time . . . . .	88

6.12	Instructions Per Cycle (IPC) of NoRecon 3D, Static, R2D3-Lite and R2D3 running different workloads. In all cases, R2D3 shows much more graceful degradation. . . . .	89
6.13	Average IPC when cluster size is increasing. The total stacked cores kept at 16 and 10000 random fault cases is tested for each point of x-axis. . . .	90
6.14	Frequency and area of R2D3 for a varying number of 3D layers in a cluster, using the proposed interconnect solution. As the number of cluster size increases, the area overhead increases due to larger crossbars and more MIVs and the frequency will decline due to increased number of layers to cross within a clock cycle, larger crossbars and more MIVs. . . .	91
7.1	Compute size achieved for 99.73% yield if no reliability solution, core-level solution, pipeline-stage level solution or module-level solution is employed for varying fault rates. Dashed line denotes the current process's failure rate of a CNTFET (10 ppm). . . . .	101
7.2	Effect of CNT distribution (varying pitch sigma) on FO4 inverter $3\sigma$ delay and yield failures for $W = 100$ nm and $s = 20$ nm . . . . .	104
7.3	a) 3DTUBE pipeline-level structure b) Parallel modular-level architecture c) Serial modular-level architecture . . . . .	107
7.4	Performance of baseline, pipeline-stage level and module level solutions of 3DTUBE for varying transistor failure rate. Dashed line denotes the current process's failure rate of a CNTFET (10 ppm). Si-based design is evaluated at a failure rate of 1 ppb. . . . .	111

## LIST OF TABLES

### **Table**

3.1	Critical system parameters for all solutions considered. 3DFAR is stacked four layers deep so, naturally, its footprint is significantly smaller. Note that it achieves almost optimal performance at much lower area and power cost than all other planar solutions. . . . .	40
6.1	The gem5 simulation parameters for R2D3. . . . .	78
6.2	Area and Power for each stage of the 5-stage pipeline. . . . .	80
6.3	Maximum temperature observed across the layers of an 8-layer design. . .	85
6.4	Feature Comparison Matrix: Fault detection and diagnosis, repair (tolerance) and prevention (lifetime management) are key features required for a reliable solution which is satisfied only by R2D3 at a low-cost. *N.R. stands for not reported . . . . .	92
7.1	Synthesis results of an OpenSPARC T1 core optimized for performance .	108
7.2	EDP reduction of pipeline-stage and module level solutions of R2D3 for an 8-core OpenSPARC T1 design at 10 ppm transistor failure rate over the Si-based design evaluated at 1 ppb failure rate. . . . .	110

## **ABSTRACT**

Over the past five decades, continued scaling of Silicon fabrication technology has led to an exponential increase in transistor budgets, leading to performance improvements. As device scaling slows down, emerging technologies such as 3D integration and carbon nanotube field-effect transistors are among the most promising solutions to increase device density and performance. These emerging technologies offer shorter interconnects, higher performance, and lower power consumption than corresponding bidimensional (2D) fabrics. However, higher levels of operating temperatures and current densities also cause an increase in wearout mechanisms that project significantly higher failure rates. Moreover, due to the infancy of the manufacturing process, high variation and defect densities, chip designers are not encouraged to consider these emerging technologies as a stand-alone replacement for Silicon-based transistors.

Conventional reliability solutions focus on one specific feature and simply assume that other requirements would be provided by different solutions. Hence, this assumption has resulted in solutions that are proposed in isolation of each other and fail to consider the overall compatibility and the implied overheads of multiple isolated solutions for one system.

The goal of this dissertation is to introduce new architectural and circuit techniques that can work around high-fault rates in the emerging 3D technologies, improving performance and reliability comparable to Silicon, while the manufacturing process is improved. Hence, this dissertation investigates the reliability problem in the high fault rate technologies and proposes opportunities to develop a low overhead and efficient solution.

This dissertation proposes a new holistic approach to the reliability problem that addresses necessary aspects of an effective solution such as detection, diagnosis, repair and prevention synergically for a practical solution. By leveraging 3D fabrics layouts, it proposes the underlying architecture to efficiently repair the system in the presence of faults. The key idea is based on a fine-grained reconfigurable pipeline for multicore processors, which minimizes routing delay among spare units of the same type by using physical layout locality and efficient interconnect switches. This thesis presents a fault detection scheme by re-executing instructions on idle identical units that distinguishes between transient and permanent faults while localizing it to the granularity of a pipeline stage.

Furthermore, with the use of a dynamic and adaptive reconfiguration policy based on activity factors and temperature variation, we propose a framework that delivers a significant improvement in lifetime management to prevent faults due to aging. Based on temperature variation across the chip and its effect on threshold voltage, the framework shows significant improvement in NBTI degradation by allowing the resources under stress to partially cool down and recover.

Finally, a design framework that can be used for large-scale chip production while mitigating yield and variation failures to bring up Carbon Nano Tube-based technology is presented. The proposed framework is capable of efficiently supporting high-variation technologies by providing protection against manufacturing defects at multiple granularities: module and pipeline-stage levels.

# **CHAPTER I**

## **Introduction**

In the past few decades, silicon technology trends into the nanometer regime have led to an exponential increase in transistor budgets, leading to drastic performance improvements. This aggressive scaling, made possible by numerous technological breakthroughs, has been the driving force behind performance and efficiency milestones in computational systems. However, as the Silicon (Si) technology approaches its fundamental limits, a variety of challenges confront it. The issues range from design nearing the power and thermal limitations to extreme process variation and wearout failures in the manufactured parts.

With the gradual slowdown of Moore's law, the semiconductor industry has seen the emergence of various new technologies to supplement or replace silicon-based transistors. As physical scaling of Si is predicted to end with the 5 nm node [1], various emerging technologies have shown great potential of supplementing silicon transistors. Hence, while chip manufacturers continue on their quest to improve CMOS technology, they are also exploring and evaluating physical devices that go beyond classic Si transistors, envisioning machines that could employ spintronics [2], optical communication [3], graphene [4], nanotubes [5], or even quantum computations [6] that are likely to suffer from even more disruptive reliability issues. More specifically, they will be forced to adopt other solutions to improve performance and density, potentially leveraging novel device technologies such as 3D integration [1] and emerging technologies which can also be used in monolithic 3D



designs, such as Carbon Nanotubes [7], by building fully complementary logic circuits [8]. In this dissertation we focus on monolithic 3D integration and Carbon Nanotube technology that are briefly introduced in the following

## **1.1 Emerging 3D Technology**

3D technology is emerging as a promising solution that can bring massive opportunities. In a 3D IC, multiple layers of logic circuits are connected using vertical vias that can vary in size from  $5\mu m$  to  $0.05\mu m$ , offering a wide range of granularity in vertical connections. 3D integration offers shorter interconnects, reduced RLC parasitics, better performance, more power savings, and a denser implementation. A vertical 3D die stack offers a higher level of integration, smaller form factor and faster design cycle.

### **1.1.1 Monolithic 3D**

Monolithic 3D (M3D) technology, by its definition, is a 3D integration technology that fabricates two or more tiers of device tiers sequentially, rather than bonding two fabricated dies together using bumps or Through Silicon Vias (TSVs). Compared with other existing 3D integration technologies (wirebonding, interposer, TSV, etc.), monolithic 3D integration is the only one that enables ultra fine-grained vertical integration of devices and interconnects, thanks to the extremely small size of inter-tier vias (shown in Figure 1.1). High-density monolithic 3D integration is enabled by using Monolithic Interlayer Vias (MIVs) can improve communication bandwidth by 10,000 times between layers compared to regular TSV-based 3D-ICs [9].

M3D integration opens up the possibility of designing cores and associated networks using multiple tiers MIVs [10] and hence, reducing the effective wire length [1]. Compared to TSV-based 3D ICs [11], M3D offers the "true" benefits of vertical dimension for system integration thanks to the extremely small size of inter-tier vias. Architects have leveraged the performance benefits of this vertical stacking to significantly decrease the

run-time of compute-intensive workloads by stacking multiple layers of cores/processors to create 3D parallel systems for data accelerators [12, 13] and Network-on-Chip (NoC) architectures [14, 15, 16, 17, 18, 19, 20, 21]. Fabrication of the first monolithic 3D IC within a foundry has been a great leap towards the realization of such architectures [22]. Recently, 64 parallel processor cores with stacked memory [23] and a large-scale 3D CMP with a cluster-based near-threshold computing architecture [24] have been demonstrated by academia. Moreover, a heterogeneous 3D FPGA (Xilinx Virtex-7 FPGA) is already in mass production [25].

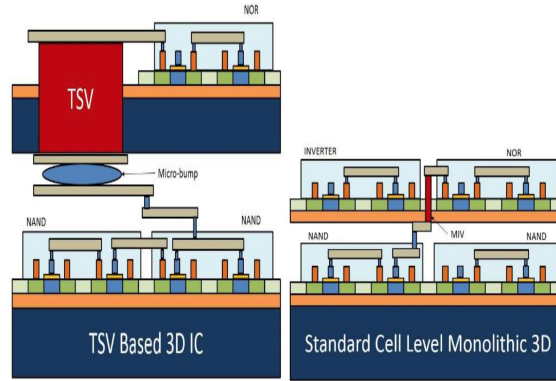


Figure 1.1: Comparison between TSV-based and monolithic 3D integration using MIVs. TSV integration adds overhead and limits the expected benefit of 3D integration. Higher integration density can be achieved through monolithic integration [26].

### 1.1.2 Carbon Nanotube Field Effect Transistors

Carbon Nanotube Field Effect Transistors (CNTFETs) are a promising alternative to Si-based devices due to the exceptional electrical, thermal, mechanical and transport properties, such as high carrier mobility, smaller gate capacitance and better current endurance [5]. Having an order of magnitude better energy-delay product (EDP) compared to conventional CMOS logic, CNTFET is a promising candidate for building energy-efficient highly integrated digital logic.

One of key challenges of dense monolithic 3D integration of logic and memory is that the processing temperatures for all upper layer circuitry must be low so as to not damage

or destroy the lower. Monolithic 3D integration of logic through the use of CNFETs due to their low processing temperature is another highly attractive aspect of this new technology [27]. Recently, CNTFETs have also been adopted to play a key role in the design and development of next-generation of 3D Monolithic System-on-a-Chip technology to create densely integrated logic and memory products [28]. In this regard, monolithic CNTFET 3D-ICs have been demonstrated by building fully-complementary logic circuits and CNTFETs on top of silicon CMOS [5].

Moreover, emerging technologies offer orders of magnitude enhancement in memory access time compared to conventional computer architectures which often assume that memory access latencies are much longer than the time spent on processing operations [29]. Massive amounts of on-chip non-volatile memory such as low-voltage resistive RAM (RRAM) [30] and magnetoresistive memories such as spin-transfer torque magnetic RAM (STT-MRAM) [31] can be integrated using Monolithic 3D-ICs, and itself can bring significant access latency and power benefits compared to off-chip storage [32]. This not only overcomes one of the most dominant bottlenecks of high performance computing, but also brings unique opportunities in re-designing various conventional micro-architectures to have more simple and efficient processors [32].

However, after decades of exploration, chip manufactures have had a good reason to procrastinate deployment of these new technologies and rely on architecture level performance enhancements in Si technology: despite the benefits that new devices and technologies offer, yield and reliability are still the major obstacles for their commercial realization. Beside the extensive research that is required on the manufacturing, architecture level innovation by rethinking of the conventional approaches can play a key role for chip production using these new technologies.

## 1.2 Reliability Challenge

Even before considering the repercussions of adopting new exotic technologies, experts envision the next integrated circuits as complex machines composed of several billions of minuscule and fragile physical devices[7]. Beyond their intrinsic fragility, the increasing device density raises both electric current and power density, phenomena which accelerate the wear of already defect-prone nanoscale transistors even further. As technology continues to evolve, the characteristics of the underlying logic components are dramatically changing: from limited and robust to plentiful and fragile. This leads to a scenario in which current architectures and design methodologies are no longer adequate.

The failure rate for electronic systems follow the *bathtub curve* [33] that describes the probability of permanent failure spanning three distinct periods: early failures (fabrication, packaging and shipping), random failures and wearout failures. Although scaling in Si technology has exaggerated all three kinds of failures, years of manufacturing research accompanied with conventional architecture and circuit level reliability solutions have made 2D Si a robust technology [34].

**Infant Period:** The beginning of the product's lifetime is characterized by an initial high rate of device failures. These high failure rates are due to latent manufacturing defects that escape the initial product testing. For 3D integration and emerging devices, the situation is worse because of the early stages of development and a higher rate of infant and random period errors [7]. The new process and use of novel materials is associated with new challenges at the beginning. Each extra manufacturing step adds a risk for defects. For example, the temperature that is required to create upper layers in a 3D circuit has been a key issue in fabricating monolithic circuit as it destroys the previously fabricated layers, creating defects and reducing reliability.

**Aging and Wear-out (Breakdown Period):** After the grace period, device failures start to occur with increasing frequency over time due to age-related wearout. Many devices will enter this phase at roughly the same time, creating an avalanche effect and a quick rise

in device failure rates. The increased transistor density and longer heat dissipation paths in 3D circuits lead to elevated levels of operating temperatures, current densities and thermal variations. Since most wearout mechanisms are highly dependent on these parameters, significantly higher failure rates are projected for future technology processes [35, 36, 37, 38, 39, 40]. Furthermore, this aggressive trend in device scaling and non-uniform temperature variation has made NBTI the most serious threat to the lifetime reliability of a chip in both present and future process technologies [41]. As a device undergoes repeated stress and recovery phases, its threshold voltage gradually spikes, causing its delay to increase over time, eventually violating the design constraints and introducing faults to the system. Neglecting these runtime hardware faults can lead to serious consequences, such as service disruption and output corruption.

**Manufacturing Process Variation:** The current CNTFET manufacturing process is riddled with imperfections [42]. Chemical synthesis of CNTs does not provide precise control over the locations of individual CNTs on the wafer and consequently, significant variations can exist in the spacing between CNTs. This non-uniformity, which is expressed as *CNT count variation*, leads directly to spatial non-uniformity in the electronic properties of CNTFETs, resulting in increased delay, signal level attenuation and failure. Moreover, a single defective CNTFET can cause faults on the gate level, such as higher leakage, less balanced rise/fall delays or too much driver strength for either pull-up/pull-down path [43]. These problems exist because the technology itself, albeit promising, is still developing from infancy.

The goal of this thesis is the exploration and propose low-cost defect-tolerance solutions for microprocessor designs that will reduce the reliability cost induced by scaling into emerging M3D and CNT or even smaller and more unreliable silicon process technologies. Thus, unless reliability concerns are addressed by effective design solutions, manufacturing yields and silicon chip lifetime expectancy will soon be drastically compromised, while future device technologies may be nonviable from the start. Extensive research on both the

manufacturing as well as architecture fronts are required to move innovation forward to create large scale chips using these new technologies. However, aggressive manufacturing research is not done unless a product using the technology is marketed but products can not be developed profitably because of the high variation in the manufacturing process. Hence, to break this causality dilemma, we as architects need to develop design flows for high variation technologies that can overcome the reliability concerns of a new technology and compete with the highly optimized state-of-the-art silicon-based processors.

### **1.3 Sustaining a disruptive technology**

As shown in Figure 1.2, for any disruptive technology to sustain growth and innovation, it usually requires a low-end market with a low barrier of entry to gain initial sources of revenue. With the initial investment, the technology can then improve to enter the mainstream markets and eventually outperform existing technologies. For example, as observed by Christensen in [44], flash memory, a disruptive technology, costed  $5\text{-}50\times$  greater than the hard disk per megabyte of memory and was not as robust for writing. However, flash chips achieve higher performance and reliability at lower power by eliminating moving parts. Flash memory started in small value networks, such as digital cameras, modems and industrial robots. Comparatively, disk drives are too big, fragile and power consuming for these applications. After flash chips succeeded as a niche, the industry began marketing specialized storage systems in portable packages, such as the thumb drive. Today, Solid State Drives (SSDs), made using flash memory, comprise the fastest growing segment of the storage, arriving to this stage because of the initial investments in low-end markets.

Hence, to sustain the development of CNT-based solutions, it is necessary to introduce architectural and circuit innovations to improve yield, such that we can produce initial designs for the low demand market use, as shown in Figure 1.2. By introducing these solutions, we will help drive the demand for 3D and CNT-based products, increasing the number of products that are manufactured. This initial income obtained can be leveraged

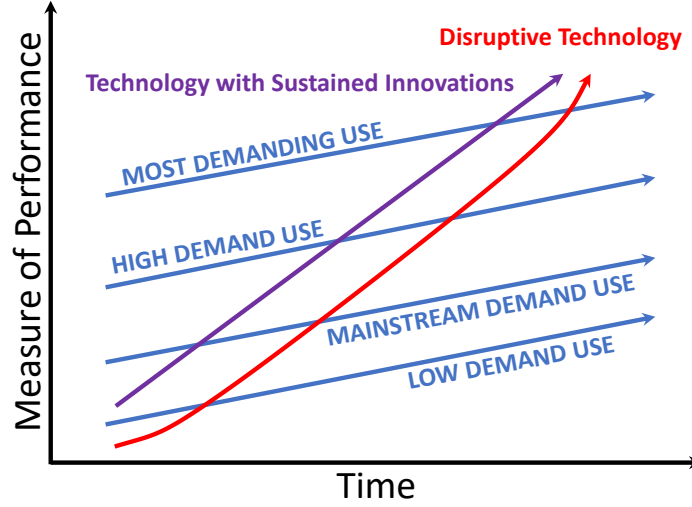


Figure 1.2: Path to sustenance for a disruptive technology [44]

by the foundry to improve the process technology and expand future lines of products, leading to a stable market.

## 1.4 Previous solutions

As described in the previous section, the failure rate for electronic systems follows what is known as the bathtub curve that describes the probability of permanent failure with three distinct periods. As shown in figure 1.3 and considering these three lifetime periods, reliability solutions can be divided into two separate categories: **Prevention** - methods that slow down aging by decelerating wearout mechanisms and **treatment** - methods that deal with the failures after they occur. Prevention methods effect wearout or failure mechanisms such as  $V_{th}$  degradation by controlling parameters like temperature, power, utilization, workload, frequency, supply voltage, etc.

Alternatively, treatment methods are designed to tolerate faults encountered during operation. A reliable treatment system requires the inclusion of three critical capabilities: mechanisms for 1) **detection** to identify the presence of a fault, 2) **diagnosis** to locate the source of the fault, i.e. to find the faulty component(s), and 3) **repair** [45]. These three characteristics from treatment systems along with prevention are the **four pillars of a com-**

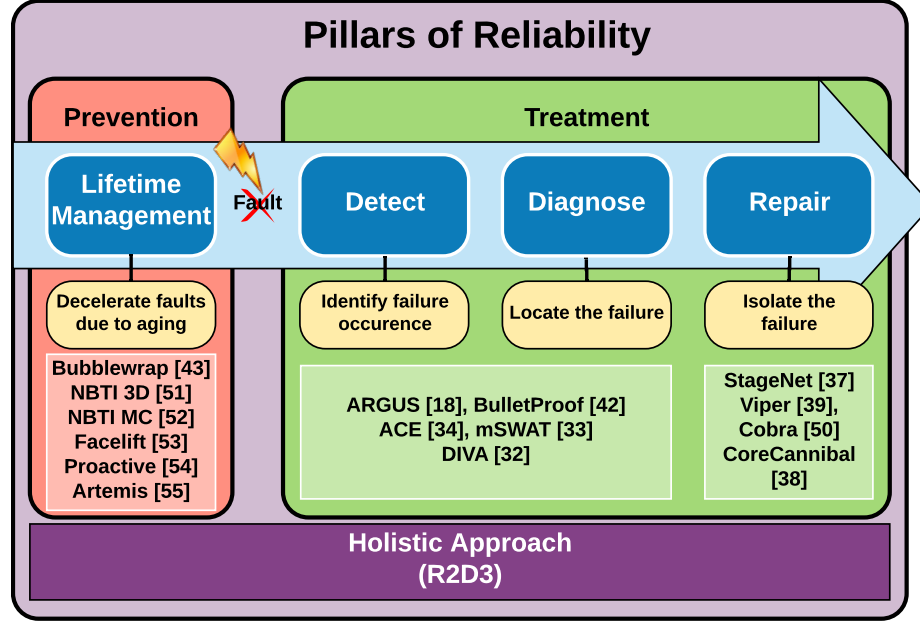


Figure 1.3: Required features for a holistic reliability solution and existing solutions built in isolation for a particular category

**prehensive solution** for reliability.

#### 1.4.1 Treatment Techniques

Prior works leverage the redundant nature of multi-core systems that allow low-cost repair by disabling defective cores [46]. However, adopting core-level mechanisms for high fault-rate technologies can cause multiple failures to discard many cores at once. Therefore, these solutions do not scale well with high fault rates [47]. Some research has focused on developing low-cost fault tolerance for classic pipelined processors that rely on online testing [48], runtime fault detection [34], or defect isolation [49, 47]. StageNet [49] has shown that fine-grained architectures can break apart the hardware units of classic hard-wired pipelines, dissolving them into a sea of redundant hardware components [50, 47, 49, 51, 52]. Upon fault, these designs can reconfigure the hardware by replacing faulty components with new ones. Bullet-Proof [48] utilizes a microarchitectural checkpointing mechanism that creates epochs of execution, during which distributed on-line built in self-test (BIST) mechanisms validate the integrity of the underlying hardware.



### 1.4.2 Lifetime Management Techniques

Traditionally, guardbanding has been used to protect against aging. Typically, operating frequency is decreased or supply voltage is increased to account for degradation to prevent timing violations due to aging. Unfortunately, guardbanding incurs a significant performance and power overhead during the entire lifetime, even though NBTI degradation does not fully accumulate until the end. Prior work have proposed several alternatives to mitigate NBTI degradation. These works include adaptive voltage scaling [53], device oversizing [54], input vector control (IVC) [55] and forward body biasing [56]. These approaches, while trying to accommodate aging, typically affect performance and power. However, they could help to push back the many-core power wall.

Aging is highly dependent on the utilization and operating temperature. System-level techniques take advantage of the application runtime behavior to improve lifetime. Adaptive voltage scaling (AVS) is an architecture-level technique proposed to mitigate aging in modern processors. Facelift [57] is a specific application of dynamic voltage scaling (DVS) in which the supply voltage only adapts once during the lifetime of a processor to switch from a slow aging mode to a high speed mode. Bubblewrap [58] uses techniques based on Facelift to enhance performance in a multi-core processor. Artemis [59] is an aging-aware application mapping and DVS scheduling framework that considers the PDN-aging of 3D NoC-based CMPs.

These works try to mitigate the effects caused by aging early on, instead of reducing it over time. The effect is limited, as when the supply voltage increases to counteract aging, the  $V_{th}$  degradation soon converges to that found in the guardbanded case [59]. This calls for an adaptive lifetime management technique deployed during runtime.

### 1.4.3 Frankenstein Solutions

While some prior work may seem to incur a lower overhead in a particular category in Table 1.3, they do not provide all the four features of reliability. Even if we ignore compati-

bility issues and combine multiple solutions together, in a Frankenstein method, the system will incur a high performance penalty and area overhead. For example, if we hypothetically want to create a combined solution with the highest coverage and best lifetime enhancement, including fault repairing, we would combine [59], [52] and [46]. It would result in a 12.5% performance and 12.8% area overhead without even considering power. Our work motivates the need for a unified solution that can concurrently provide all the features of reliability at a low overhead.

#### 1.4.4 Shortcomings of previous solutions

Challenges associated with low yield and high fault rates of the 3D technology call for the incorporation of both prevention and treatment mechanisms in one solution. Although there is a plethora of work to address reliability concerns, no work addresses all the four features of reliability concurrently. The issues with previous solutions can be summarized as follows:

**First**, previous solutions, as has been included in Figure 1.3, focus on one specific pillar of reliability and provide remedy for that issue. Some solutions lack essential features to be considered practical. For instance, previous reconfigurable architectures such as StageNet [49], Core Cannibal architecture [50], and Viper [47] lack fault detection and diagnosis at a finer granularity which makes them incomplete.

**Second**, studies shows that elevated temperatures and longer heat dissipation paths in 3D ICs lead to rapid aging and higher fault rates [35, 36] and this calls for an end-to-end solution that can control aggressive aging and repair the system upon a fault. Many of these solutions are designed for low fault rates of Si technology and do not scale well in the higher fault rates.

The **third** issue with the previous approaches is that most of these solutions are proposed for a particular architecture and are not compatible with each other, so it is impossible or extremely difficult to combine them to create a holistic solution.

The **fourth**, some of these solutions are too complicated and incur too much overhead on performance, area or power consumption. For instance, previous reconfigurable architectures such as StageNet rely on buffered switches with limited bandwidth between consecutive stages. These switches in each pipeline stages boundary complicate the design as different instructions may take different number of clock cycle to reach to the destination and a more complicated mechanisms and control infrastructure for ensuring correct execution. More importantly, these extra buffers between stages introduce performance overhead in terms of IPC as well. Consequently, writers in StageNet paper had to change and redesign the structure of the 5-stage in-order pipeline to create a 4-stage processor and reduce the number of signals and amount of data between each two stages. The performance overhead and cost of redesigning the whole processor makes the adaptation of these architectures for more complicated architectures such as out-of-order cores or those with more number of stages too expensive.

And the **last** is the high-overhead *Frankenstein* issue: while some prior work may seem to incur a lower overhead in a particular category in Table 6.4, they do not provide all the four features of reliability. Even if we ignore compatibility issues and combine multiple solutions together, in a Frankenstein method, the system will incur a high performance penalty and area overhead. For instance, StageNet [49], core cannibal architecture [50], and viper [47] lack fault detection and diagnosis at a fine-granularity which is essential to their repair system. This introduces additional overhead on the top what they propose as fault detection and diagnosis at a fine granularity can be expensive [48, 34]. Moreover, the mentioned solutions miss-out on considering usage and heat leading to non-uniform aging, and hence, lower lifetime and higher faults.

## 1.5 Contributions of this thesis

While the use of either prevention or treatment mechanisms in prior solutions to provide reliability may sufficed for current technologies, dealing with challenges of low yield and

high fault rates technologies calls for the incorporation of both simultaneously. Moreover, these solutions handle faults (prevention or treatment) on the core level which only handles a limited number of faults and is suitable for standard technologies like Silicon. With the increasing defect rate, a single device failure cause entire cores to be discarded, often times with the majority of the components of the faulty core still functional. These solutions do not scale well with high fault rates. This motivates a rethinking of the architectural fabric from the ground up, with dynamic adaptivity and configurability as primary requirements. To be effective, the fault isolation has to be at a granularity finer than core-level. This dissertation investigates the reliability problem in the high fault rate in 3D technologies and introduces the challenges and opportunities to develop a low overhead and efficient solution by making the following contributions.

### **1.5.1 3DFAR: A New fabric for Reliability of Processors in 3D**

We propose 3DFAR, a novel reliability solution for multi-core processor designs, which leverages the system’s natural redundancy to provide robustness to permanent transistor failures. We exploit the spatial locality of equivalent compute units to design efficient interconnect switches. These switches enable dynamic sparing of equivalent units upon the occurrence of a fault. They are extremely low in area footprint and they present minimal propagation delay, both because of their innovative design and because of the much shorter distances to traverse vertically in reaching the spare unit, compared to a bi-dimensional layout. Our experimental evaluation indicates that 3DFAR outperforms several state-of-the-art solutions, when implemented with area-equivalent resources. In summary, we make the following contributions:

- A novel sparing-based, reliable solution for multicore processors, specialized for 3D fabrics. Our solution entails minimal performance impact over an unprotected 2D design (<5%).
- A new method to connect corresponding hardware resources on a vertical layout,

which does not require any buffering or complex routing. Through our method, we can dynamically create and adapt pipelines of healthy resources.

- An analysis of the proposed interconnect solutions and their performance when varying the number of 3D design layers.

### 1.5.2 R2D3: A Holistic Solution

We propose **Reliability by Reconfiguring 3D systems – R2D3**, a holistic, aging-aware reliability engine with fine-grained reconfigurability for parallel streaming systems that can concurrently **detect, diagnose, repair and prevent** failures at runtime. The engine comprising of a controller, reconfigurable crossbars and detection circuitry, takes advantage of the smaller routing delay over vertical layers in 3D circuits to create a fast-reconfigurable fabric that leverages the availability of identical resources in parallel systems. Our controller creates epochs of execution, at the end of which the detection circuitry can access the inputs and outputs of all layers to verify correctness and diagnose faults. The R2D3 controller manages the reconfigurable fabric to concurrently provide single-cycle replay detection and diagnosis, fault-mitigating repair and aging-aware lifetime management during runtime. Furthermore, to prevent faults, the controller considers temperature variation across the chip and activity factor of each stage to adaptively reconfigure functioning stages to create virtual pipelines. This balances out the aging rate, extending the lifetime of the system.

R2D3 can be adopted to parallel architectures that have large number of identical units, such as systolic arrays, mesh-based systems, many-core or multi-core systems. In this work, we study the effects of R2D3, for vertically-stacked, in-order OpenSPARC T1 pipelines, as such pipelines closely resemble the basic compute units in modern-day streaming accelerators and massively parallel systems [60, 61]. In summary, this paper makes the following contributions:

1. Demonstrate a practical reliability engine, R2D3, for high fault rate monolithic 3D

technology systems that can concurrently support prevention, detection, diagnosis and repair during runtime.

2. Propose a mechanism as part of R2D3 to detect and diagnose faults by re-executing instructions on idle units. It detects and distinguishes transient and permanent faults using single-cycle replay and localizes faults at the granularity of a pipeline unit.
3. R2D3 prevents failures by introducing graceful degradation using smart scheduling policies at the stage-level. It considers the activity factor and temperature variation of each unit to adaptively create virtual pipelines that balance out the aging rate.
4. A new solution which incorporates the thermal map of processing cores and the 3D system characteristics to mitigate concentrated usage and minimize  $V_{th}$  degradation. This policy does not add any performance overhead to the underlying reconfigurable system.

### **1.5.3 3DTUBE: A Design Framework for High-Variation Carbon Nanotube-based Transistor Technology**

This section proposes a design flow framework that can be used for large-scale chip production while mitigating yield and variation failures to bring up CNT-based technology, using a reliable reconfigurable architecture. The proposed framework is capable of efficiently supporting high-variation technologies by providing protection against manufacturing defects at multiple granularities: module and pipeline-stage levels. To achieve this goal, we first develop a flexible variation model based on CNT density fluctuations, that allows the designer to mimic the yield obtained for different manufacturing processes. Leveraging the fact that CNTFETs can be used in CMOS logic, we built a 16 nm CNTFET standard cell library and characterized for voltages varying from 0.4 V to 0.7 V to build a design library that can be used to synthesize a logical design. Furthermore, to enable the commercialization of CNTFET-based products, along with the use of the variation model

and the design library, we propose the use of a 3D multi-granular reconfigurable architecture, **3DTUBE**, to improve yield and throughput of these high-variation designs.

**Dissertation Organization:** The remainder of the dissertations is organized as follows. In Chapter 2, a background on some required basic is discussed. Chapter 3 presents 3DFAR which is a fine-grained solution to repair a 3D system in presence of faults. Chapters 4, 5 and 6 discuss R2D3 which is a holistic solution that addresses all aspects of a reliable system. Chapter 7 introduces our solution for reliability of CNTFET-based 3D technology which is called 3DTUBE and Chapter 8 is the concluding section to this dissertation.

## CHAPTER II

### Background

#### 2.1 Lifetime Failure Models

This section presents an overview of the major aging and wearout effects that impact lifetime significantly such as time-dependent dielectric breakdown (TDDB) [62], negative bias temperature instability (NBTI) [63] and electromigration (EM) [64]. In fact, most wearout mechanisms exhibit an exponential dependence on temperature [65, 66].

The RAMP [67] approach models the mean time to failure (MTTF) of a processor as a function of temperature related failure rates of individual structures on chip and proposes to calculate the reliability by applying the sum-of-failure-rates (SOFR) model. Alternatively, Vattikonda *et al.* present a divide-and-conquer-based reliability evaluation that employs a Monte Carlo based simulation and argue that their results are more accurate and realistic [68]. As the heart of reliability evaluation in this work is based on Monte Carlo simulations, we adopted the latter strategy.

The various aging and wearout mechanisms can be modeled as follows:

##### 1) Negative Bias Temperature Instability (NBTI)

Bias temperature instability is a degradation phenomenon for MOSFETs. Its highest impact is observed in PFETs when stressed with negative gate voltages at elevated temperatures. NBTI degrades many essential transistor parameters like: (1) threshold voltage; (2) transconductance; (3) sub-threshold slope; (4) channel mobility; (5) saturation, drain, and



off currents; and (6) delay. In order to account for static and dynamic NBTI degradation, a device-level prediction model was proposed by Vattikonda *et. al.* [69], demonstrating the accuracy of this model against the results obtained from physical experiments. The model used to compute the change in threshold voltage caused by NBTI degradation for stress and recovery phases are as follows:

Stress:

$$\Delta V_{th} = \sqrt{K_v^2 (t - t_0)^{1/2} + \Delta V_{th0}^2} \quad (2.1)$$

Recovery:

$$\Delta V_{th} = V_{th} [1 - \sqrt{\eta(t - t_0)/t}] \quad (2.2)$$

where,

$$K_v = A \cdot t_{ox} \cdot \sqrt{C_{ox}(V_{gs} - V_{th})} \cdot [1 - V_{ds}/\alpha(V_{gs} - V_{th})] \cdot e^{(E_{ox}/E_0)} \cdot e^{-(E_a/kT)} \quad (2.3)$$

where,  $\Delta V_{th}$  is the change in threshold voltage caused by NBTI stress and recovery phases,  $t_0$  is the starting time of the stress or recovery phase,  $\Delta V_{th0}$  is the threshold voltage change starting each stress or recovery phase at  $t_0$  and  $K_v$  is a constant that lumps all the technological parameters [69].

The effect of NBTI on the lifetime of a device can be expressed using mean time to failure (MTTF) by NBTI. Schroder and Babcock [70] and Wu *et. al.* [71] modeled the MTTF caused by NBTI using the following equation:

$$MTTF_{NBTI} = A_{NBTI} (1/V_{gs})^\gamma \cdot \exp(E_{aNBTI}/kT) \quad (2.4)$$

where  $V_{gs}$  is the gate-to-source voltage,  $A_{NBTI}$  is a technology related constant,  $\gamma$  is the voltage acceleration factor and  $E_{aNBTI}$  is the activation energy for NBTI.

## 2) Time-Dependent Dielectric Breakdown (TDDB)

Thin gate oxides lead to additional failure modes as devices become subject to gate oxide wear-out (e.g., Time Dependent Dielectric Breakdown, TDDB). Over time, gate oxides can break down and become conductive. If enough material in the gate breaks down, a conduction path can form from the transistor gate to the substrate, essentially shorting the transistor and rendering it useless[33]. Fast clocks, high temperatures, and voltage scaling limitations are well-established architectural trends that aggravate this failure mode.

The model for  $MTTF_{TDDB}$  is described by the following expression[62]:

$$MTTF_{TDDB} \propto (1/A)^{1/\beta} (F)^{1/\beta} V_{gs}^{a+bT} . e^{(\frac{c}{T} + \frac{d}{T^2})} \quad (2.5)$$

where  $A$  is the transistor's gate oxide area,  $\beta$  is the Weibull slope parameter,  $F$  is cumulative failure percentile.

### 3) Electromigration (EM)

Due to the momentum transfer between the current-carrying electrons and the host metal lattice, ions in the conductor can move in the direction of the electron current. This ion movement is called electromigration. Gradually, this ion movement can cause clustered vacancies that can grow into voids. These voids can eventually grow until they block the current flow in the conductor. This leads to increased resistance and propagation delay, which in turn leads to possible device failure. Other effects of electromigration are fractures and shorts in the interconnect. The trend of increasing current densities in future technologies increases the severity of electromigration, leading to a higher probability of observing open and short-circuit nodes over time. The accepted model for MTTF due to electromigration is based on Black's equation and is as follows[68]:

$$MTTF_{EM} \propto (J - J_{crit})^{-n} . e^{\frac{E_{aEM}}{kT}} \quad (2.6)$$

where  $J$  is the current density in the wire,  $J_{crit}$  is the critical current density required for electromigration,  $E_{aEM}$  is the activation energy for electromigration.

#### **4) Hot Carrier Degradation (HCD)**

As carriers move along the channel of a MOSFET and experience impact ionization near the drain end of the device, it is possible to gain sufficient kinetic energy to be injected into the gate oxide. This phenomenon is called Hot Carrier Injection. Hot carriers can degrade the gate dielectric, causing shifts in threshold voltage and eventually device failure. HCD is predicted to worsen for future thinner oxide and shorter channel lengths.

Of all wearout mechanisms listed above, NBTI has been shown to be the dominant factor to affect the lifetime of a system [41]. Hence, we optimize our adaptive reconfigurable lifetime management policy for NBTI-based aging.

## **2.2 3D Circuits and Challenges**

3D integration technology is actively being studied as a solution to continue the scaling trajectory predicted by Moore's Law. In a 3D IC, multiple layers of logic circuits are connected using vertical vias that can vary in size from 5 $\mu$ m (Through Silicon Vias, TSVs) to 0.05 $\mu$ m (Monolithic Inter-Tier Vias, MIVs), offering a wide range of granularity in vertical connections. 3D integration offers Shorter interconnects, reduced RLC parasitics, better performance, more power savings, and a denser implementation. A vertical 3D die stack offers a higher level of integration, smaller form factor and faster design cycle. TSV is the wafer-to-wafer bonding. Wafers are processed separately, and they are stacked and connected after each wafer is fabricated. The advantage of TSV is that the upper-layer process does not have the temperature constraint so that conventional Si process can be used. The disadvantage of TSV is that due to the wafer and via alignment issue, the via to via distance is limited to  $\mu$ m level, which results in low via density and coarse 3D integration [72, 73].

Monolithic 3D technology, by its definition, is a 3D integration technology that fabricates two or more tiers of device tiers sequentially, rather than bonding two fabricated dies together using bumps or TSVs. Compared with other existing 3D integration technologies (wirebonding, interposer, TSV, etc.), monolithic 3D integration is the only one that enables

ultra fine-grained vertical integration of devices and interconnects, thanks to the extremely small size of inter-tier vias (typically 50nm in diameter). High-density monolithic 3D integration is enabled by using Monolithic Inter-layer Vias(MIVs) with the same size as a regular local via [74] (diameter < 100 nm) and can improve communication bandwidth by 10,000 times between layers compared to regular TSV-based 3D-ICs [9]. However, the disadvantage of the monolithic 3D integration is that the upper layer fabrication process has a temperature constraint. Since the upper-layer process should not affect the lower layer's back end, the process temperature is limited to  $400^{\circ}\text{C}$  [75] which is far lower than the Si CMOS process temperature, which is  $1100^{\circ}\text{C}$  [76]. Low temperature process technologies such as a carbon nanotube transistor, resistive random access memory, or Ge nanowire have a potential for monolithic 3D integration.

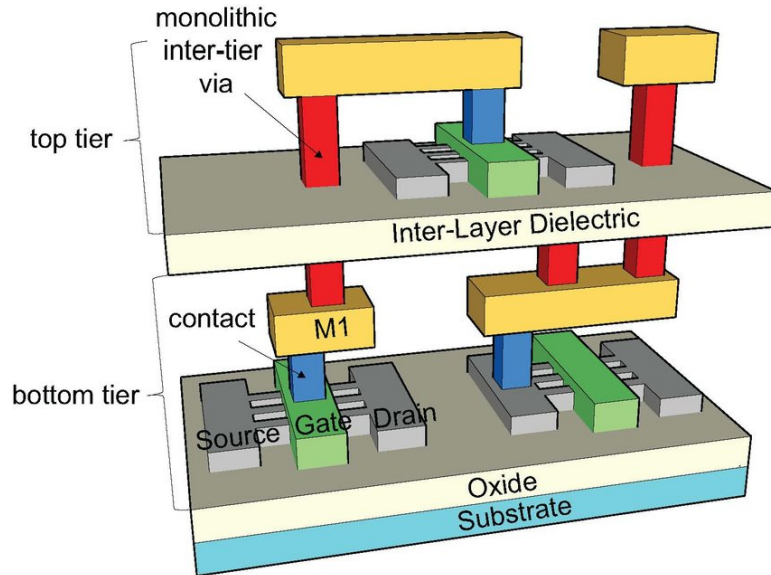


Figure 2.1: Monolithic 3D integration using MIVs [77]

Recently, 64 parallel processor cores with stacked memory [23] and a large-scale 3D CMP with a cluster-based near-threshold computing architecture [24] have been demonstrated by academia. Moreover, a heterogeneous 3D FPGA (Xilinx Virtex-7 FPGA) is already in mass production [25].

### 2.2.1 CNT Technology

Carbon-based devices leverage the carbon hexagonal lattice structure. Graphene and graphene nanoribbon (GNR) are two-dimensional structures, and Schottky barrier type carbon nanotube field effect transistors (SB-type CNFETs) and MOS-type CNFETs are one-dimensional nanowire structures. Graphene, the two-dimensional hexagonal lattice structure of carbon, was first investigated because of its very high mobility ( $120,000 \text{ cm}^2/Vs$  theoretically,  $200 \text{ cm}^2/Vs$  for after transistor fabrication of small-width graphene nanoribbons) [78] and high cut-off frequency ( $400 \text{ GHz}$ ) [79]. Despite these advantages, graphene has high off-state current since it has zero bandgap. Because of this, graphene cannot be used as a channel material for field effect transistors. Instead, carbon nanotube and graphene nanoribbons, which have high mobility and non-zero bandgap, are used as a channel material for field effect transistors.

MOS-type CNFETs are type of CNFETs, whose source and drain contacts are Ohmic. The current flow mechanism of MOS-type CNFETs is the thermionic emission modulation, wherein the channel energy barrier controls the source to drain current flow such as the usual Si MOSFETs. The possible application of CNFETs is in low power electronics because of its high mobility and gate-all-around structure. High mobility makes CNFETs have low latency with the same dynamic power dissipation ( $\tau = \frac{CV}{I_{on}}$ ,  $P = CV^2$ ). The gate-all-around structure enhances the gate controllability to channel potential, which results in steep on-off switching and low leakage power dissipation. CNFETs have been shown to be excellent candidates for low voltage and near threshold operations making them perfect candidates to be used in the design of sensors, IoT devices and energy-constrained devices.

Moreover, the low process temperature makes it possible for CNFETs to be used in monolithic 3D integration [76, 80]. The high thermal conductivity helps CNFETs mitigate the power burden of 3D integration. The advantage of MOS-type CNFETs is that they have better off-state current blocking and higher on-state current conductance because there is no Schottky barrier in the source and the drain contact [81, 82]. This means MOS-Type

CNFETs have lower leakage current, higher on-state current, and lower Subthreshold.

The disadvantage of MOS-type CNFETs over SB-type CNFETs is that the Ohmic contact is harder to fabricate than the Schottky contact. In case of CNFETs with doped CNT Ohmic contacts, the doping control in nanoscale devices is very difficult. This causes dopant number and position fluctuation that results in transistor performance variation [81]. In case of CNFETs with Ohmic metallic contacts, because the metal work function is fixed, it is difficult to fabricate. Also, n-type MOS-type CNFETs have p-type SB behavior, which can increase the leakage current [83]. As a result, SB-type CNFETs are preferred for large scale digital integrated circuits [82].

Recently, carbon nanotube field-effect transistors (CNFETs) have been of great interest due to their better electrical, thermal, mechanical and transport properties [84, 85]. Having an order of magnitude better energy-delay product (EDP) compared to conventional CMOS logic [86], they are promising candidates for building energy-efficient highly integrated digital logic [29, 87].

One of key challenges of dense monolithic 3D integration of logic and memory is that the processing temperatures for all upper layer circuitry must be low so as to not damage or destroy the lower. Monolithic 3D integration of logic through the use of CNFETs due to their low processing temperature is another highly attractive aspect of this new technology [27]. In this regard, monolithic CNFET 3D-ICs have been demonstrated by building fully-complementary logic circuits [84] and CNFETs on top of silicon CMOS [85].

### **2.2.2 Systems Using M3D and CNT Technology**

N3XT [29] is an experimental demonstration of advances in robust CNFET technology and on-chip non-volatile memory, enabled by fine-grained monolithic 3-D integration and cooling solutions. N3XT is an example on how the next generation of massively integrated electronic systems can revolutionize massive data computing and can deliver unprecedented computing capacity in a highly energy-efficient manner [88]. Massive amounts

of on-chip non-volatile memory such as low-voltage resistive RAM (RRAM) [30] and magnetoresistive memories such as spin-transfer torque magnetic RAM (STT-MRAM) [31] can be integrated using Monolithic 3D-ICs, and itself can bring significant access latency and power benefits compared to off-chip storage [32].

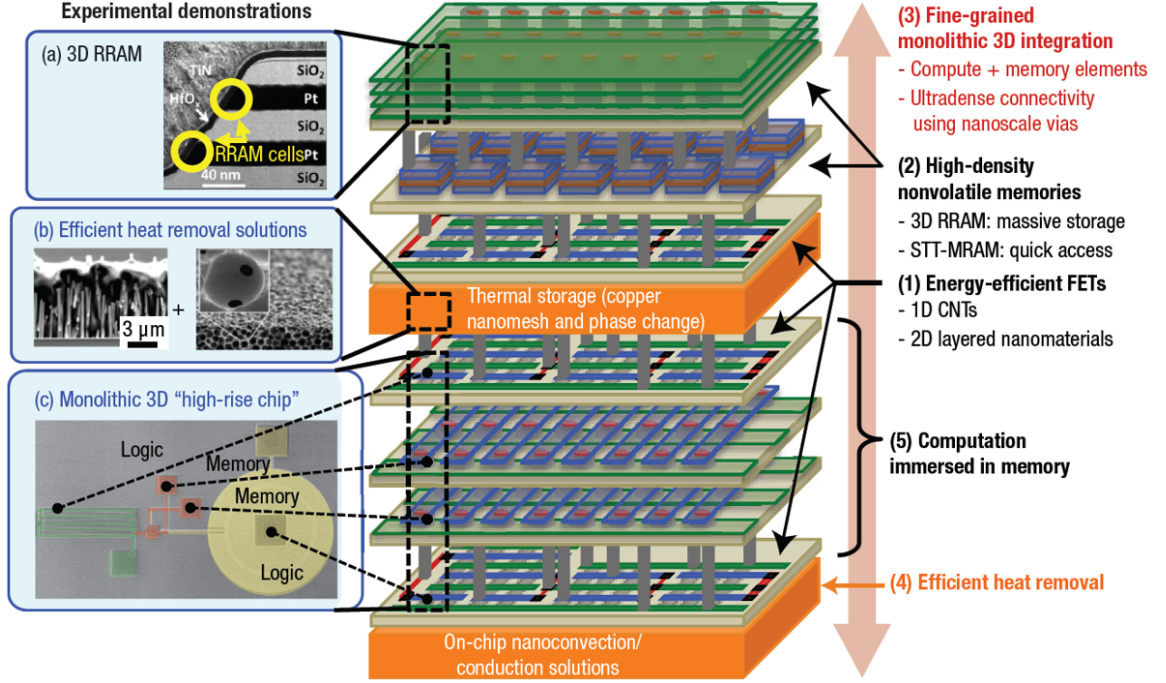


Figure 2.2: Monolithically integrated 3D system for abundant data computing [80].

Moreover, emerging technologies offer orders of magnitude enhancement in memory access time compared to conventional computer architectures which often assume that memory access latencies are much longer than the time spent on processing operations [29] and have complex memory hierarchy structure to overcome these limitations. This not only overcomes one of the most dominant bottlenecks of high performance computing, but also brings unique opportunities in re-designing various conventional micro-architectures to have more simple and efficient processors [32]. It was shown that CNT sensors can be integrated together with the data processing unit and memory, and this on-chip three-dimensional integration provides high bandwidth, low EDP, and *in-situ* data processing. This research shows that integration of multiple emerging devices can improve the data col-

lection (3D integrated CNT sensor), data transporting (on chip monolithic 3D integration of sensor and processing unit), data storage (3D integrated RRAM), and data processing (monolithically integrated CNFET and 2D materials) significantly.

These new technologies can bring opportunities to one other important aspect of integrated systems that is reliability. These new technologies and aggressive device scaling have greatly exacerbated the occurrences and impact of faults in computing systems, which has made ‘reliability’ a first-order design constraint [89]. The heterogeneous soft error resilience characteristic of 3D integration’s vertical stacking structure is due to its shielding capability. 3D-layered structures allow outer-layers to shield inner-layers from particle strikes [90]. Moreover, vulnerable instructions can be mapped to robust layers and the previously proposed reliability-hardening techniques can be selectively applied on microarchitecture components residing in vulnerable layers. Thus, the hardware complexity and the cost for achieving the same reliability target can be minimized [89]. Zhang et al. [89] characterize soft error vulnerabilities across the stacked layers under 3D integration technology and show that the outer-dies can shield more than 90% particle strikes for the inner-dies, which leads to a heterogeneous error rate across layers in a 3D chip.

So by considering the natural resiliency of 3D fabric against soft errors, permanent faults and failures due to wearout and manufacturing defects would be first concern in designing the next generation of processors. Specially, by considering that 3D integration and the new devices technologies making it possible are new and immature and would take years to reach the reliability of traditional CMOS technology.

So we in this work, we are showing that not only the new 3D-fabric leads to benefits like shielding capability, footprint and wirelength reduction, it also brings new chances by creating microarchitectural and architectural opportunities to overcome new reliability concerns.



## CHAPTER III

# **A Three-Dimensional Fabric for Reliable Multi-Core Processors**

In the past decade, silicon technology trends into the nanometer regime have led to significantly higher transistor failure rates. Moreover, these trends are expected to exacerbate with future silicon devices. To enhance reliability, a number of approaches have been proposed that leverage the inherent core-level and processor-level redundancy present in large chip multiprocessors. However, all of these methods incur high overheads, making them impractical.

In this chapter, we propose 3DFAR, a novel architecture leveraging 3-dimensional fabrics layouts to efficiently enhance reliability in the presence of faults. The key idea is based on a fine-grained reconfigurable pipeline for multicore processors, which minimizes routing delay among spare units of the same type by using physical layout locality and efficient interconnect switches, distributed over multiple vertical layers. Our evaluation, based on performance measurements on a physical design, shows that 3DFAR outperforms state-of-the-art reliable 2D solutions, at a minimal area cost of only 7% over an unprotected design. This section explains the 3DFAR architecture and the mechanisms that help it repair and tolerate faults. The proposed fault detection and diagnosis policy is introduced.

### 3.1 Introduction

While deep sub-micron technology enables the placement of billions of transistors on a single chip, it also poses unique challenges: processors in aggressively scaled technologies are more susceptible to permanent transistor failures at runtime, often due to wearout phenomena. Moreover, based on predictions by the ITRS [1], soon it will no longer be economically viable for companies to continue to shrink transistors' dimensions. Instead, chip manufacturers will be forced to turn to other solutions to boost performance, possibly novel device technologies that are likely to suffer from even more disruptive reliability issues. Thus, unless reliability concerns are addressed by effective design solutions, manufacturing yields and silicon chip lifetime expectancy will soon be drastically compromised, while future device technologies may be nonviable from the start.

Today, a number of solutions exist to address reliability and fault tolerance in processors; they can be grouped into software management solutions and hardware level techniques. Software management methods usually present high latency in fault detection and slow performance in recovery. On the other hand, hardware approaches are often based on providing spare units or using inherent redundancy within a time domain or a sharing infrastructure. However, approaches that provide spare components, such as N-modular redundancy methods [91], DIVA [92], StageNet [49], BulletProof [48], *etc.*, are associated with hardware and power overheads, which can be quite considerable in the case of a many-core processor.

In this chapter, we leverage recent advances in 3D integration to address these issues and propose a reliable and efficient layout structure and design solution. As device scaling based on Moore's Law slows down, 3D integration appears to be one of the most promising solutions to continue to increase design density and performance. Although 3D integration presents challenges of its own, including heat dissipation and lack of specialized design tools, they are gaining increasing attention because of their shorter interconnects, higher performance, lower cost, and lower power consumption than corresponding 2D fabrics.

Our solution, called a 3-Dimensional FABric for Reliable multicores (3DFAR), proposes to use monolithic 3D fabrics to stack corresponding hardware units from distinct cores above each other, and leverages inter-core redundancy to provide a reliable architecture. We place equivalent resources within short vertical distance from each other, and provide low overhead and fast communication infrastructure using MIVs.

In our architecture, illustrated in Figure 3.1, we replace the direct connections at each pipeline stage boundary by a crossbar switch, so that each stage may connect to subsequent stages from other layers. By adaptively routing around failed stages we can salvage working units and performance effectively. In developing our solution, we investigated a range of interconnect switch structures, and studied their area overhead and performance. We evaluated our solution on the physical design of a 4-core in-order processor, running 12 distinct test programs and compared it against several state-of-the-art 2D reliable architectures. We found that 3DFAR provides consistent performance improvement over these solutions, at a minimal area cost of only 7% for interconnects and MIVs. In summary, we make the following contributions:

- A novel sparing-based, reliable solution for multicore processors, specialized for 3D fabrics. Our solution entails minimal performance impact over an unprotected 2D design ( $<5\%$ ).
- A new method to connect corresponding hardware resources on a vertical layout, which does not require any buffering or complex routing. Through our method, we can dynamically create and adapt pipelines of healthy resources.
- An analysis of the proposed interconnect solutions and their performance when varying the number of 3D design layers.

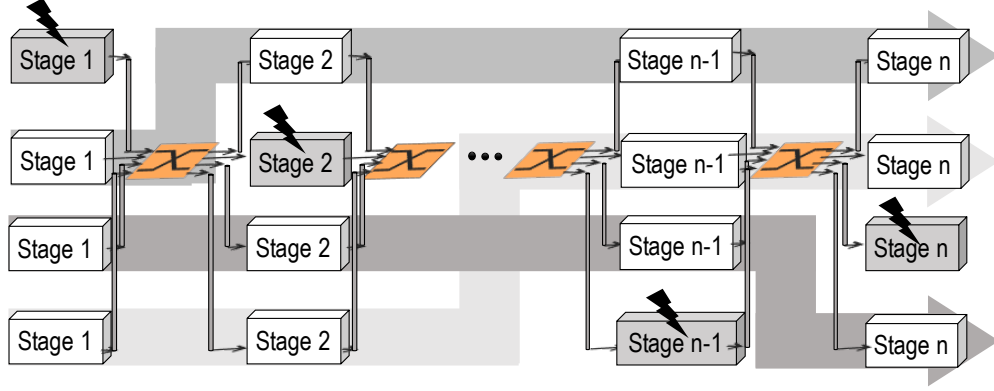


Figure 3.1: Schematic of the 3DFAR approach. In 3DFAR multi-core architectures, corresponding pipeline stages are stacked vertically, while specialized crossbar units are inserted between each pair of stages. In a four-faults situation as the one illustrated, a regular 2D CMP would be completely disabled. In contrast, 3DFAR can dynamically reconfigure to connect healthy units as shown with the wideband lines, providing the computing power of 3 complete cores.

### 3.2 Architecture

3DFAR is a novel reliability solution for multi-core processor designs, which leverages the system's natural redundancy to provide robustness against permanent transistor failures. It can be dynamically configured to route instructions through functioning hardware components and detour around failed pipeline stages. Unlike classic architectures that execute instructions on paths fixed at design time, 3DFAR relies on inter-stage crossbar switches to form logical pipelines dynamically. As illustrated in Figure 3.1, by replacing the direct connections at each pipeline stage boundary with interconnect switches, we create a network of resources in which each pipeline stage is connected to all instances of the subsequent stage. To minimize the performance loss from inter-stage communications, we use multiplexer-based full crossbar switches because of its non-blocking access to all of their inputs, and small number of inputs and outputs that are not prohibitively expensive. The mux-based crossbar has a fixed channel width and, as a result, the delay of transferring an instruction from one stage to the next can take place with a small frequency overhead within the same clock cycle of pipeline stage execution when implemented in 3D.

By adaptively routing around failed stages we can salvage working units and effectively

repair the system. When a fault occurs, the victim unit (*i.e.* a pipeline stage) is isolated, and an identical unit from another core, laid out on a different layer of the 3D fabric, is used to advance the execution. Hence, the pipeline executing the application may comprise elements from various vertical layers, connected together to form a logical processor core. With reference to the example in Figure 3.1, where 4 faults have disabled units on each vertical layer, 3DFAR can build 3 complete pipelines dynamically, as shown by the wideband lines, while a traditional solution (2D or 3D) would be completely disabled.

3DFAR’s cross-layer interconnect switches do not require any buffering, thus greatly simplifying their design and control requirements. Moreover, it is possible to use switches to connect pipeline stages both on forward and backward paths. Since propagation delays on vertical MIVs are minimal (approximately two orders of magnitude faster than in conventional 2D layouts) due to the much shorter lengths to be traversed, we can avoid buffering by accommodating a small increase in clock period.

As illustrated in Figure 3.1, 3DFAR provides a network of resources in which each pipeline stage is connected to all instances of the subsequent stage using multiplexer-based crossbar switches. By adaptively routing around failed stages, working units can be salvaged in the network to improve performance effectively.

### 3.3 Latency of Interconnect Switches

3DFAR cross-layer interconnect switches do not require any buffering, thus greatly simplifying their design and control requirements. Moreover it is possible to use switches to connect pipeline stages both on forward and backward paths. Note that prior solutions, *e.g.*, [49], suffer from performance and complexity impacts introduced by buffered switches. However, since propagation delays on vertical MIVs are minimal (approximately two orders of magnitude faster than in conventional 2D layouts) due to the much shorter lengths to be traversed, we can avoid buffering by accommodating a small increase in clock cycle length ( $<5\%$ ).

To this end we pursued an analysis of propagation delays for two distinct layouts of a 4-core processor. The first is a 2D layout with the 4 cores in a 2x2 formation. Switches to replace individual faulty pipeline stages with healthy units from other cores are placed at the center of the formation. The second is a 3D layout with the 4 cores stacked above each other, similarly to the schematic of Figure 3.1. The cores implement a subset of the Alpha instruction set architecture and are mapped to 45nm IBM technology. The processor implements an in-order pipeline with control and data forwarding paths and integrated data and instructions local cache. We referred to the data in [11] to calculate propagation delays through MIVs. We then used SPICE to measure the worst-case propagation delay for signals going from the output of one stage, through an interconnect switch (a simple multiplexer for the study) and to the input of the next stage. Figure 3.2 reports our findings: the 2D layout presents a propagation delay of 950ps, compared to only 50ps for the 3D layout, a 95% reduction. This vast difference is due to the much shorter distances that must be traversed to reach a corresponding unit in the three-dimensional solution. Based on these findings, we developed novel interconnect switch designs, which minimize delay, while balancing the silicon footprint of each design layer. Our designs are detailed in Section 3.4.

### **3.3.1 Flexible Deployment of 3DFAR**

Because of its low latency, which does not require buffering, it is straightforward to deploy 3DFAR in a wide range of stacked processor architectures. In this context, Figure 3.3 provides an example of crossbar switches inserted in a 5-stage in-order pipeline with data forwarding paths, as the one used for our propagation delay analysis: 5 switches are introduced to advance computation between pipeline stages, 3 are used for data and control forwarding connections, and one more switch connects the memory stage to the integrated local cache.

Finally, with the 3DFAR solution it would also be possible to time-multiplex resources,

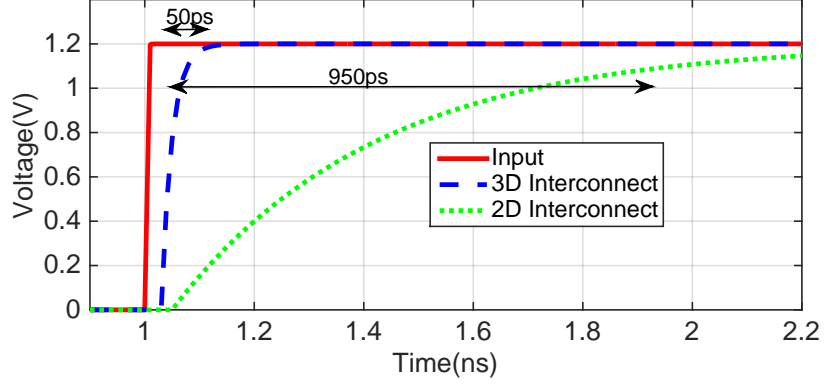


Figure 3.2: Propagation delays through inter-stage interconnects for a 4-core planar layout vs. a 3D stacked layout. The large delay for the planar design is mostly due to the much longer wires required to connect corresponding units.

thus pushing further out the availability of the system. Note that this extension requires additional hardware support and the introduction of buffering capabilities for at least some of the pipeline stages.

### 3.3.2 Number of Design Layers

There are contrasting goals in determining the ideal number of layers in a 3DFAR design: on one hand the more the layers the more spare units are available, and thus the stronger the robustness of the solution. On the other hand a high number of design layers may be impractical and may negatively affect the latency required for traversing the vertical dimension of the design to reach a spare unit. We study this trade-off in the experimental evaluation section.

## 3.4 Interconnect Switch Design

In designing our interconnect switch architecture, we took into account the number of vertical MIVs required for each interconnect, the propagation delay entailed, which in turn affects the nominal operating frequency of the system, and the silicon area overhead for each silicon layer. Note that the overall area overhead is the one of the largest silicon layer.

The number of vertical MIVs necessary to connect all vertical pipeline stages depends

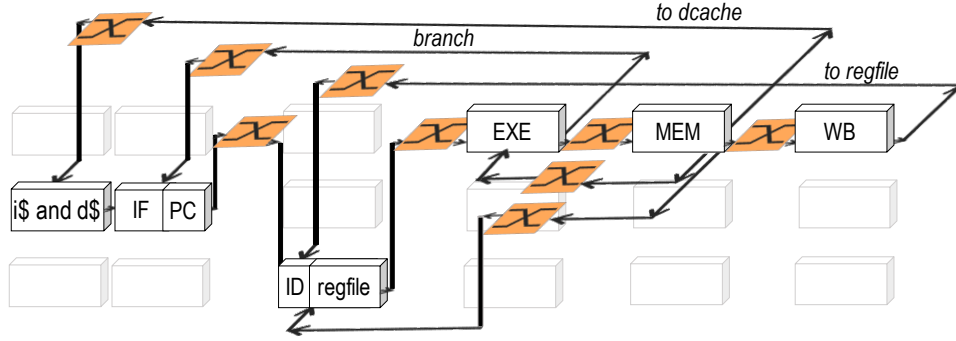


Figure 3.3: Middle-layer and vertically distributed interconnect switches. a) The middle-layer solution entails the worst area overhead, concentrated in the middle layers. Delay overhead does not exceed  $2 \times \text{traverse\_delay}(\# \text{layers} / 2)$ . b) The vertically distributed design balances area at the switch granularity, delay overhead can be as much as the cost of traversing all layers.

on the specific architecture of the system. For the microarchitecture used in our examples and depicted in Figure 3.3, a total of 1,106 signals must be connected to and from other layers. Each pipeline stage uses a varying number of input signals, ranging from 68 for the connection between write-back stage and the register file in the decode stage, to 336 signals connecting decode to execute. The propagation delay between connections depends on how many vertical layers a signal must cross to go from a source pipeline stage through the crossbar and then to the destination stage. Finally, the silicon area overhead was estimated based on the size of individual MIVs, as reported in [11] ( $0.5\mu\text{m}$  TSV) and the area of the crossbar switch. In light of these factors, we developed three design solutions, described in the sections below.

Finally, an important factor associated with using MIVs is their reliability, as the failure of a single MIV may cause unpredictable effects that could lead to system failure. Yield and reliability improvements are usually achieved through a range of redundancy techniques and sparing, investigated in recent years, along with several diagnosis and repair mechanisms [93].



### 3.4.1 Middle-Layer Interconnect

The goal of this interconnect solution, outlined on the left side of Figure 3.4, is to minimize and equalize the latency introduced by the interconnect switches. By placing all the switches in the middle layer, all signals travel no more than  $2 \times \text{traverse\_delay}(\# \text{layers} / 2)$  to go from one layer, through the switch and to the destination layer. When the number of layers is even, switches can be placed on the two middle layers in an alternating fashion, without affecting the overall latency impact. Note that the middle layer must accommodate MIVs incoming from all other layers of the design. However, they can be aligned so that the same surface can be used for MIVs coming from layers above and thus coming from layers below. Thus, with this solution, the middle layer requires space for  $1,106 \cdot n / 2$  MIVs, where  $n$  is the number of layers. In addition, we estimated the area of each crossbar (for 1 bit) to be approximately half the area required by MIVs. Thus, in first approximation, this solution requires an area overhead equivalent to  $1659 \cdot n / 2$  MIV-equivalent area units. If the number of layers is even, crossbar switches can be partitioned over two layers, and the area cost is reduced to  $1382 \cdot n / 2$  MIV-equivalent area units.

### 3.4.2 Vertically Distributed Interconnect

To balance the area overhead experienced by the vertical layers of the design, we explored a vertically distributed solution, where each interconnect switch is placed on a different layer, on a rotating fashion. This approach minimizes area imbalance, at the granularity of one switch. Note, however, that the number of signals connecting two stages of a pipeline varies for each stage; thus each switch entails a different area overhead, depending on how many signals it must route. The right side of Figure 3.4 illustrates this solution.

With this approach, the interconnects located at the bottom and top layers experience the longest wire delays, that is, the time to traverse all vertical layers, so they are best placed at the input of pipeline stages which can afford more timing slack. The area overhead is highly dependent on the specific switch locations: for each layer, it includes the area of the

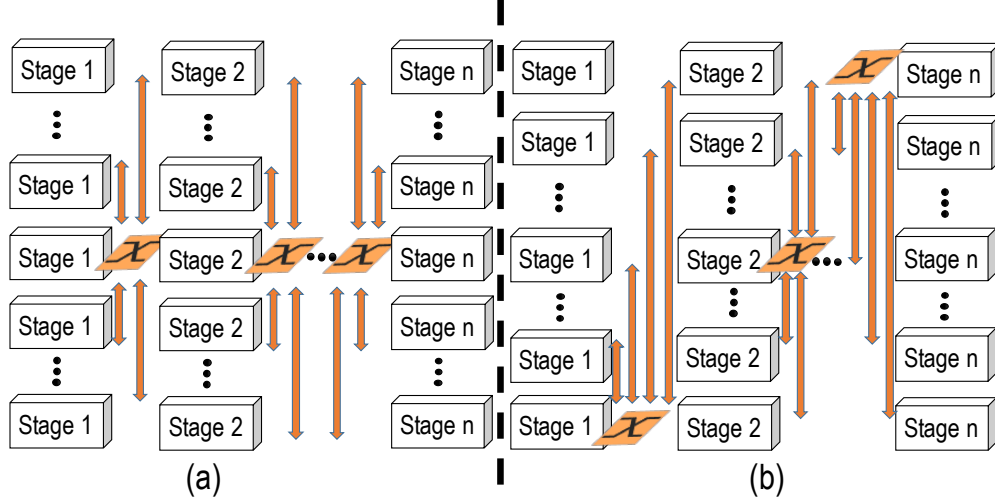


Figure 3.4: Middle-layer and vertically distributed interconnect switches. a) The middle-layer solution entails the worst area overhead, concentrated in the middle layers. Delay overhead does not exceed  $2 \times \text{traverse\_delay}(\# \text{layers} / 2)$ . b) The vertically distributed design balances area at the switch granularity, delay overhead can be as much as the cost of traversing all layers.

local switch(es), the area of the MIVs passing through to reach switches on other layers, and the areas of the MIVs incoming to the local switch, from above or below (which can be overlapped). In general, middle layers host the most MIVs, approximately  $1,106 \cdot n / 2$ , ( $n$  being the number of layers) and likely one or a few switches. Thus the area overhead is generally slightly less than in the prior solution.

### 3.4.3 Vertical Bus Interconnect

This solution leverages a bus-style architecture, where vertical links are run across the entire height of the design, and each layer uses a set of multiplexers to select its inputs among one of the vertical bus lines or the prior stage on the same layer, as shown in Figure 3.5.a). The advantage of this solution is that only unidirectional MIVs are required, since signals are switched directly at their destination layer. Note that the propagation delay for this solution depends on the location of the faulty unit: signals must simply propagate from a faulty layer to their spare unit, without the need to reach a fixed-location switch. In the worst case scenario, this delay is still equivalent to crossing the entire set of design layers,

as for the vertically-distributed interconnect switch.

The area overhead is uniformly distributed among all layers, indeed each layer is simply augmented with a set of selector multiplexers. Moreover, the number of vertical signals to be routed is half than in previous solutions. We estimated the area of the selector multiplexers to be approximately  $1/4$  that of an MIV, for each signal. Thus, with a vertical bus structure, each layer must accommodate  $(553+138) \cdot b = 691 \cdot b$  MIV-equivalent area units, where  $b$  is the number of vertical buses between each stage. This is the most balanced and minimal area solution among all those that we explored.

Finally, note that we do not need to route as many vertical bus lines as the number of layers in the design. In fact, we can leverage the observation that as more units in a stage become faulty, there will be a decreased need to transfer data to healthy units in the subsequent stage. With reference to the schematic in Figure 3.5.b) , if only one unit in stage  $i$  is faulty, then we need at one vertical bus line to connect from some other layer to the corresponding healthy stage  $j$  on the faulty-unit layer . Similarly, if all but one stage  $i$  are faulty, we need at most one vertical bus to connect the remaining healthy stage  $i$  forward. The most demanding scenario occurs when half of the stages  $i$  and half of the  $j$ s are faulty, and the faults are all on distinct layers: in this case we do need indeed  $\lfloor \#layers/2 \rfloor$  vertical buses. Based on the analyses provided above, this latter interconnect entails the least area overhead at no extra cost in latency.

### 3.5 3DFAR System-Level Operation

The 3DFAR architecture, deploying one of the interconnect switch designs discussed above, is capable of replacing any pipeline stage with a spare from another layer. We assume that the control inputs of the crossbar switches are connected to a few register bits, which in turn can be programmed via the 3DFAR firmware routine. The firmware routine is called by the operating system each time a new fault is detected with information on which unit has failed. It then computes how many dynamic pipelines can be setup with the

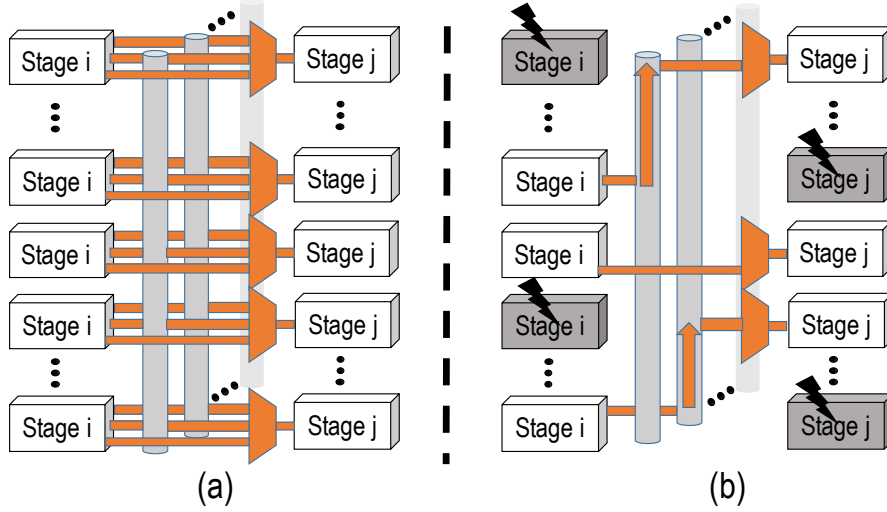


Figure 3.5: The vertical bus interconnect a) uses a set of multiplexers in each layer to select which input to route to a pipeline stage, from one of the vertical busses or from the same layer. b) Example illustrating that no more than  $\leq \lfloor \#layers/2 \rfloor$  vertical busses are needed.

current failure map, and programs the interconnect switches accordingly. It is possible that, as the result of a fault, fewer processes may execute concurrently on a system than it was possible before the fault occurrence. It is then necessary to swap the status of the victim process out to memory. This activity requires to i) flush the local cache(s) of the process, so that memory data is properly updated, and ii) save program counter, register file and status register to the memory reserved for the context of the victim process.

### 3.5.1 Faulty Registers and Register Files

To address situations where faults occur in register storage, which is critical in preserving the status of a process, we equip each critical register with a few ECC bits [94]. After the occurrence of the first fault in one of these units, we tag the unit as failed. However, thanks to the correction capabilities of ECC, we can still retrieve the data stored in the unit before disabling it. In addition, register files have typically at least two read ports: thus, even after one fault occurrence, there is at least one available port to read registers values and transfer them to memory.

### 3.5.2 Faulty Local Cache Units

Augmenting cache lines with a few additional ECC bits is fairly common practice today, even for level-1 caches [94]. When a fault hits a cache unit, it is easily detected by the ECC checksum, and simultaneously repaired. Data in cache lines affected by a fault is then relocated and the cache line is marked faulty. In the context of 3DFAR, we need to maintain access to the cache, even in presence of faults that may affect the read ports to the cache. We solve this problem by disabling a cache if faults have disabled all but the last read port of a local cache. In our example, the local cache has at least two read ports, since it is an integrated data and instruction cache. For caches with a single read port, it is necessary to duplicate the port logic, so to have a spare in case of a fault hitting the only working port.

Once all processes affected by a fault are swapped out, the 3DFAR firmware reconfigures the system to attain a complete set of working pipelines, and then the operating system may reschedule processes to execute on the newly formed pipelines.

## 3.6 Experimental Evaluation

To evaluate 3DFAR we deployed it on a 4-cores, 5-stage in-order pipeline implementing a subset of the Alpha instruction set architecture. The design is a widely-used instructional processor for advanced computer architecture courses. We augmented it with the vertical bus interconnect solution discussed in Section 3.4.3, synthesized it on an IBM 45nm technology with Synposys' Design Compiler, and placed and routed it with Cadence's Encounter. To create the 3D layout we followed the specifications and design rules recommended in [11], and evaluated power and timing through SPICE simulations. The layout obtained is presented in Figure 3.6. We also considered three baseline designs: *unprotected 2D* is the 4-core processor with no reliability protection, laid out in a bidimensional, 2x2 matrix formation. *2D w/switches* is augmented with interconnect switches placed at the center of the matrix to minimize wire lengths, as recommended in [49]. Because of the

high latency impact of the interconnect, we also implemented a buffered interconnect solution, *StageNet*, as specified in [49]. Note that buffering has a significant impact on overall system performance, particularly in the case of branch mispredictions. Finally, all systems were evaluated by executing a suite of 12 test programs, implementing a range of parallel algorithms, including search, sort, multiplication, fibonacci, *etc.*, overall executing for approximately 100,000 dynamic instructions. Table 3.1 reports our measurements for critical system parameters in absence of faults, for all the solutions described. It can be noted that 3DFAR provides the same average clock cycles per instructions (CPI) as the unprotected 2D design, although its operating frequency is 4.1% lower at 714Mhz. In contrast, the 2D unbuffered switches solution suffers from significant clock frequency slow down, while StageNet has a 39% worse CPI than 3DFAR, compounded with an additional slow down in clock frequency.

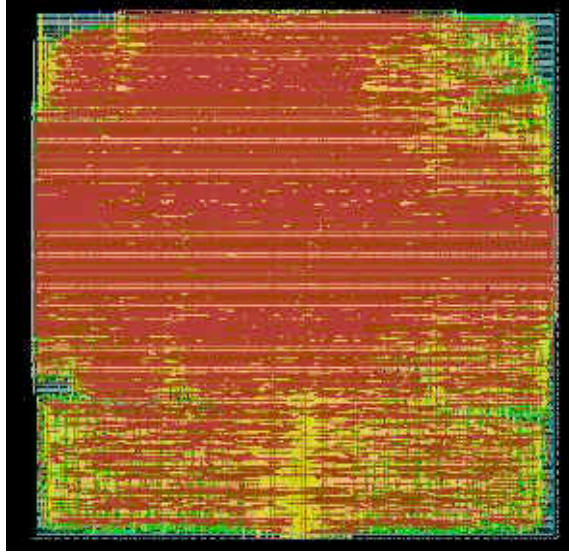


Figure 3.6: Layout of one 3D layer including a complete core (no cache) and all vertical bus switches.

### 3.6.1 Fault Model

Our fault model injects permanent transistor failures into any design component and any layer, proportionally to the area of the unit. Once a pipeline unit is hit by a fault, we disable

Layer	1 (MHz)	2 ( $\mu m^2$ )	3 area	4 (mW)	CPI
unprotected 2D	745	160,000	0%	201	0.402
2D w/switches	434	161,217	12%	222	0.402
StageNet	691	161,992	19%	274	0.561
3DFAR	714	41,234	7%	204	0.402

Table 3.1: Critical system parameters for all solutions considered. 3DFAR is stacked four layers deep so, naturally, its footprint is significantly smaller. Note that it achieves almost optimal performance at much lower area and power cost than all other planar solutions.

the entire unit and trigger a dynamic reconfiguration via the 3DFAR firmware. As discussed in Section 3.5, faults in local caches are handled with a finer granularity approach. If a fault hits an interconnect switch, we disable the unit connected to the output of that switch. If a fault hits the pipeline’s control logic, we disable the entire pipeline. We assume that MIVs are implemented reliably as discussed in Section 3.4: in Table 3.1 we accounted for one spare MIV every 100, based on the recommendation in [93]. To attain statistical confidence, we repeated each experiment on faulty processors 10,000 times, using a different random seed in injecting faults.

### 3.7 Performance in Presence of Faults

Next, we compared the robustness of 3DFAR against a number of recent reliability solutions in Figure 3.7; specifically: *Viper* [47], *StageNet* [49], *BulletProof* [48] and our basic design with no reliability protection (*unprotected 2D*). The plot evaluates the performance of the solution in instructions-per-cycle (IPC) for a wide range of faults in the system, up to 1,000 concurrent faults. To compare these different solutions, we assumed area-equivalent implementations of each solution, by considering a budget of 2B transistors, similarly to the analysis in Figure 9 of [47]. With this budget, one could implement 128 *unprotected 2D* cores, 27 *BulletProof* pipelines, 30 *StageNet* pipelines (the latter two having a fault-free throughput equivalent to about four in-order cores), enough units to build 40 *Viper* virtual pipelines, or 22 *3DFAR clusters*, each 8 layers deep. We used a depth of 8 because with

this setup, interconnect switches still allow better performance than StageNet, the crossing point being 10 layers.

3DFAR provides better performance than all other solutions at all rates beyond 38 faults. The unprotected design has the best performance at 0 faults, but quickly degrades to the worst option at 298 faults. StageNet and BulletProof provide a valuable performance degradation beyond 300 faults, while Viper shows a sustained improvement over both. Note how the compact area footprint and the limited latency cost of 3DFAR deliver a significant IPC boost even over Viper. This advantage, however, starts to thin out beyond 800 faults. We believe this is due to the benefits of the Viper’s decentralized control logic, which provides enhanced reconfiguration flexibility. On the other hand, 3DFAR’s approach is orthogonal to Viper, and the two solutions could be easily integrated.

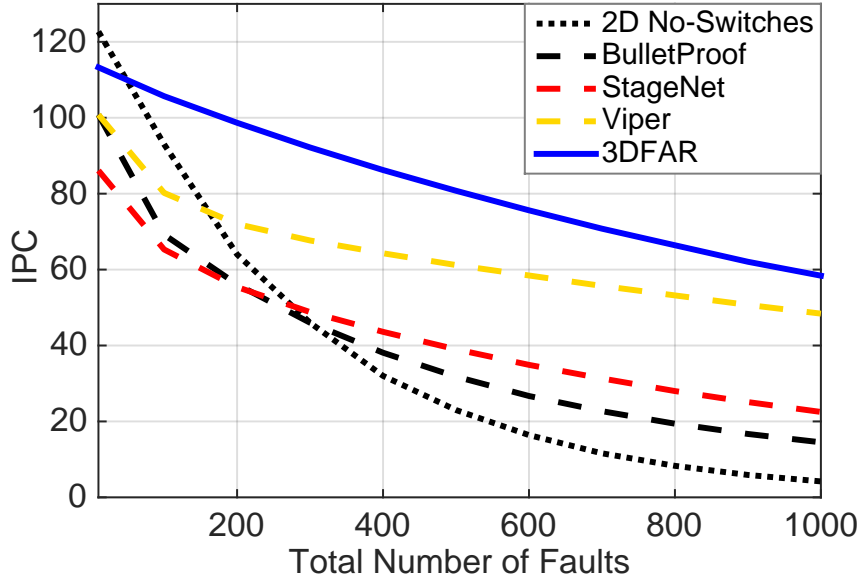


Figure 3.7: Performance of 3DFAR against state-of-the-art solutions and a baseline 2D design over varying fault numbers.

### 3.7.1 3DFAR Layers Depth

In this section we analyze the impact of increasing the number of layers on area footprint and propagation delay of the interconnect switches, which in turn affects system’s frequency. To ascertain the maximum number of layers that can be efficiently stacked to-



gether to form a cluster, we evaluated reliability and overhead over varying cluster sizes. Increasing cluster size improves reliability as there will be more available spare units, but it negatively affects area footprint and interconnect's propagation delay, which in turn impacts system's frequency.

The first important factor on determining the cluster size is the wire the delay and area overhead caused by crossbars and vertical connections. In Figure 3.8, we plot these two metrics with respect to the number of layers in the 3DFAR system. Note that the area footprint we measure is that of the largest layer in the design. We analyzed this aspect for all three interconnect switch designs proposed, to validate our prior analysis suggesting that the vertical bus interconnect provides the best area and latency solution points. Figure 3.8 reports our findings for all three interconnect solutions discussed, showing that vertical bus switches provide the best performance. As the number of stacked cores increases, the area overhead will increase because of the larger crossbars and more vertical lines required to connect the stages in these cores vertically. The working frequency is decreasing as the number of stacked cores increases because of the delay overhead of the larger crossbars and farther cores in up and bottom of the stacked structure which have to cross more layers to communicate.

To evaluate the gain from increasing the cluster size, we made an experiment by changing the cluster size and a fixed number of stacked cores. Figure 3.9 shows the IPC for different cluster sizes when the total stacked cores kept 16. The experiment shows increase in reliability benefits by increasing the cluster size. However, the returns diminish with the increasing number of coupled cores beyond 8-10 pipelines which has a marginal impact. This is so because, as a cluster spans more and more slices, the variation in time to failure of its components gets smaller and smaller. Thus, in a larger set of stages, most fail in the stages with the most area and failure rate.

One way to reduce this overhead, is to reduce the number of vertical MIV lines. As mentioned in the last section, to ensure the reliability scenarios of fault for  $n$  stacked cores,

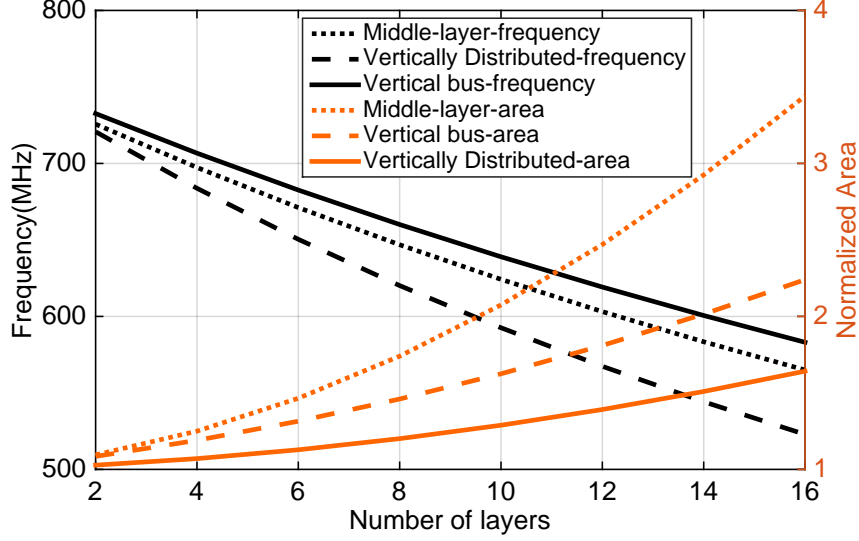


Figure 3.8: Frequency and area of 3DFAR for a varying number of 3D layers, using the proposed interconnect solutions. As the number of stacked cores increases, the area overhead increases due to larger crossbars and more MIVs and the frequency will decrease as the total frequency will decline due to larger crossbars and more MIVs.

there are  $n/2$  vertical MIV buses required and reducing them will cause failure in some of fault scenarios. But these cases might be rare that keeping some of these MIV buses only for them may not worth it. Figure 3.10 shows the average IPC for different total MIV buses, when the total stacked cores and the cluster size is kept at 16 and 10000 random fault scenarios are tested exhaustively for each point of x-axis. Based on this experiment, reducing total vertical buses is causing less than 2% and 6% in performance for less than 30 total fault injected although the gain in frequency and area will be considerable. So based on the required trade-off between reliability and performance the total vertical buses can be reduced in our proposed method.

### 3.8 Vertical Connections Reliability

Low MIV and TSV yield and reliability are serious problems, as the failure of a single MIV may cause unpredictable effects that could eventually lead to system failure. Yield and reliability improvements are usually achieved by different forms of redundancy. MIV yield

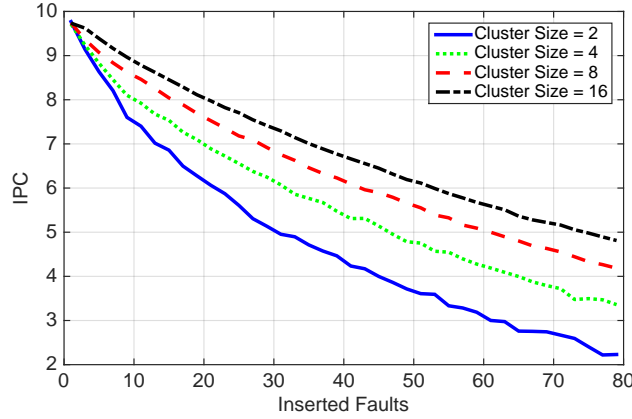


Figure 3.9: Average IPC when cluster size is increasing. The total stacked cores kept at 16 and 10000 random fault cases is tested exhaustively for each point of x-axis.

improvement using spares has been widely investigated and many diagnosis and repair mechanisms has been proposed in this regard [95, 96, 93].

The addition of spare MIVs to repair faulty functional MIVs is an effective method for yield and reliability enhancement, but this approach results in hardware cost and delay overhead. In the double MIV technique, each faulty MIV(f-MIV) is paired with an additional spare MIV(s-MIV) to enhance yield and reliability [93]. In the fault-free scenario, a signal is transferred through the two MIVs simultaneously. Once an f-MIV becomes faulty, there is still an sMIV to pass the signal. Since the two MIVs are used to pass the same signal, no additional control circuits are required. However, due to the significant area overhead induced by s-MIVs, this technique is impractical.

To reduce area overhead, the “shared s-MIV” technique has been proposed. In this technique, a set of f-MIVs is partitioned into several groups, and single or multiple s-MIV(s) are subsequently assigned to each group. By inserting MUXes and carefully designing the reconfigurable routing paths, the s-MIV(s) can be used to pass signal(s) in the presence of defective f-MIVs. Figure 3.11 illustrates an example of a shared s-MIV structure. When all the f-MIVs are fault-free, the signals are transferred by them in Figure 3.11.a. Once an f-MIV fails, the signal corresponding to the faulty one has to be shifted, which causes

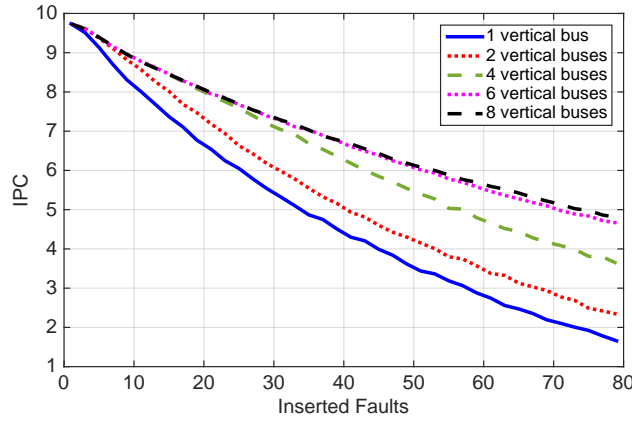


Figure 3.10: Average IPC for different total MIV buses. The total stacked cores kept and the cluster size is kept at 16 and 10000 random fault cases is tested exhaustively for each point of x-axis.

all signals between the faulty MIV and the sMIV to be shifted, as shown in Figure 3.11.b. Although this method reduces the area overhead compared to the double MIV technique, additional delay is introduced due to signal re-routing and the extra components.

Figure 3.12 shows the average number of working vertical buses for different number of added spares per each 100 MIVs. This figures show that by adding 1 or 2 % to the number of MIVs, we are able to decrease the redundancy considerably. What also needs to be considered is that the whole crossbars and vertical area overhead is only 7% of the entire design and the total area overhead due to spare MIVs is negligible. Also the reader should remember that the inserted fault number in figure 3.12 only belongs to faults in crossbar structure. So if we consider the equal failure probability for crossbar connections and logical units, 10 fault in the crossbar structure entails 140 faults in total processor as its area is only 7% of the entire design.

As mentioned before, there are several identical vertical connections that different cores are sharing which gives this architecture a natural redundancy. Moreover, several spare vertical connections can be added to each vertical connection to insure the level of resiliency required for this architecture. As high density monolithic 3D integration has been possible

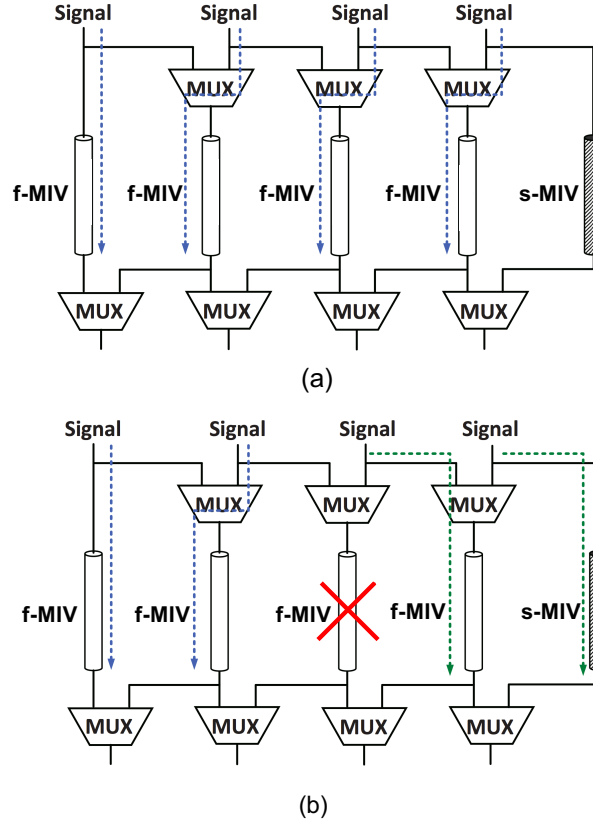


Figure 3.11: A conceptual figure on how spare MIV methods work (a) Operation when all f-MIVs are fault-free (b) Operation when the third f-MIV is faulty.

yet and it is in a developing phase and reliability aspects of vertical connection yet has to be explored.

### 3.9 Related Work

A number of recent works that strive to provide processor reliability have focused on unit sparing, exploiting natural redundancy in VLIW cores [48], introducing logic to enable dynamic reconfiguration around faulty pipeline stages [49, 47], or sparing at the core level [97]. All these solutions assume or introduce an underlying fault detection mechanism, including traditional BIST techniques and software-based fault detection [97]. Similarly, 3DFAR also assumes the presence of a fault detection mechanism and provides an efficient technique to isolate the faulty unit and repair the system around it. Among these prior

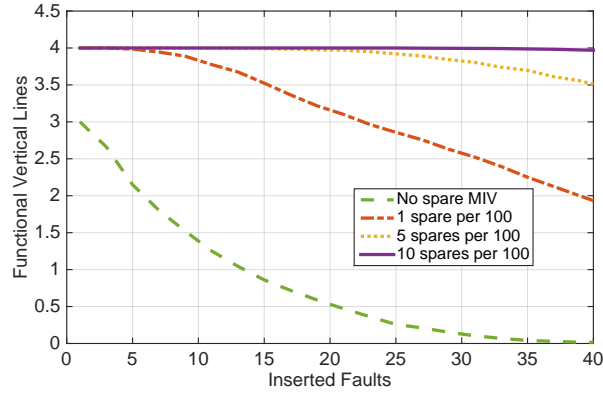


Figure 3.12: Average number of working vertical buses for different number of added spare MIVs per each 100s for different number of inserted faults in vertical connection structure for a 4-core structure.

works, StageNet [49] is the most similar to 3DFAR conceptually; however, our solution provides a much more efficient unit-isolation mechanism, which leverages the 3D layout and a novel and efficient interconnect switch. Viper [47] differs from the solutions above in that it also entails a completely distributed control logic solution, based on a service-oriented execution paradigm. However, this approach is affected by a number of limitations typical of distributed control architectures and, as a result, its performance and scalability compares poorly against traditional chip multi-processors. Note that 3DFAR is a complementary solution to Viper, and we believe it would be possible to build a 3D-stacked Viper solution using 3DFAR’s interconnect switches.

Research in reliability leveraging 3D layouts has also been explored. The authors of [98] investigate the concurrent execution of a program on two separate layers in a 3D design for fault detection, by using idle resources in the second layer. The authors of [99] propose a DIVA [92] checker processor stacked vertically over a main processor, recommending to use an older technology node for the checker processor, so to attain further robustness. In the context of reliability for 3D layouts, 3DFAR provides a complete recovery solution with extremely graceful performance degradation, compared to the limited and specialized approaches mentioned above.

### 3.10 Summary

In this chapter, we presented 3DFAR, a novel reliability solution for multi-core processor designs, which leverages the system’s natural redundancy to provide robustness to permanent transistor failures. We exploit the spatial locality of equivalent compute units to design efficient interconnect switches. These switches enable dynamic sparing of equivalent units upon the occurrence of a fault. They are extremely low in area footprint and they present minimal propagation delay, both because of their innovative design and because of the much shorter distances to traverse vertically in reaching the spare unit, compared to a bi-dimensional layout. Our experimental evaluation indicates that 3DFAR outperforms several state-of-the-art solutions in this space, when implemented with area-equivalent resources. For instance, at 200 faults, a 160-cores equivalent 3DFAR solution outperforms all other area-equivalent solutions by at least 36%. In absence of faults, 3DFAR is only 4% slower than an unprotected 2D design, and it outperforms StageNet by over 40%. Finally, the area overhead of 3DFAR is only 7% beyond that of an unprotected design at 4-stacked layers.

## CHAPTER IV

# **R2D3: A Holistic Solution for Reliability of 3D Parallel Systems**

Monolithic 3D technology is emerging as a promising solution that can bring massive opportunities, but the gains can be hindered due to the reliability issues exaggerated by high temperature. Conventional reliability solutions focus on one specific feature and assume that the other required features would be provided by different solutions. Hence, this assumption has resulted in solutions that are proposed in isolation of each other and fail to consider the overall compatibility and the implied overheads of multiple isolated solutions for one system.

This chapter proposes a holistic reliability management engine, R2D3, for post-Moore's M3D parallel systems that have low yield and high failure rate. The proposed engine, comprising of a controller, reconfigurable crossbars and detection circuitry, provides concurrent single-replay detection and diagnosis, fault-mitigating repair and aging-aware lifetime management at runtime. This holistic view enables us to create a solution that is highly effective while achieving a low overhead.

This section explains the R2D3 architecture and the mechanisms that help it repair and tolerate faults. The proposed fault detection and diagnosis policy is introduced.



## 4.1 Introduction

With the decline of Moore’s law, M3D integration opens up the possibility of designing cores and associated networks using multiple tiers by utilizing monolithic MIVs [10] and hence, reducing the effective wire length [1]. Compared to TSV-based 3D ICs [11], M3D offers the ”true” benefits of vertical dimension for system integration thanks to the extremely small size of inter-tier vias. Architects have leveraged the performance benefits of this vertical stacking to significantly decrease the run-time of compute-intensive workloads by stacking multiple layers of cores/processors to create 3D parallel systems for data accelerators [12, 13] and NoC architectures [14, 15, 16, 17, 18, 19, 20, 21]. Fabrication of the first monolithic 3D IC within a foundry has been a great leap towards the realization of such architectures [22]. However, despite the benefits offered, **yield** and **reliability** are still the major obstacles for commercial realization of this technology [35, 36, 100, 101]. Processors in 3D technology are more susceptible to permanent transistor failures, often due to wearout phenomena such as time-dependent dielectric breakdown (TDDB) [62], negative bias temperature instability (NBTI) [63] and electromigration (EM) [64]. Recent studies have shown how the elevated temperatures and longer heat dissipation paths in 3D ICs lead to significantly rapid aging and higher fault rates [35, 36, 37, 38, 39, 40]. Hence, to commercialize this new technology, new architectural and circuit modifications that can work around high-fault rates are required, improving performance comparable to Silicon, while the manufacturing process is perfected.

As shown in Figure 4.1, previous solutions can be divided into two categories: **prevention** – methods that slow down aging by decelerating wearout; and **treatment** – methods that deal with the failures once they have occurred. Prevention methods affect wearout or failure mechanisms, such as  $V_{th}$  degradation, by controlling parameters like temperature, power, utilization, workload, frequency and supply voltage. Treatment methods are designed to tolerate faults encountered during operation. As shown in Figure 4.1, a reliable treatment system requires the inclusion of three critical capabilities: 1) **detection** to iden-

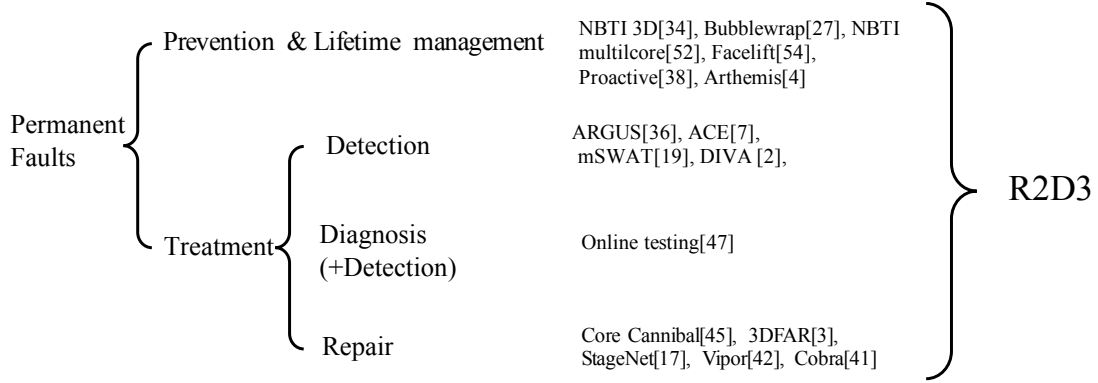


Figure 4.1: Required features for holistic reliability solution and existing solutions are built in isolation for a particular category

tify the presence of a fault, 2) **diagnosis** to locate the source of the fault, i.e. to find the faulty component(s), and 3) **repair** to isolate the failure from the system [45]. These three characteristics of treatment, combined with prevention, are the **four pillars of a holistic reliability solution**.

Challenges associated with low yield and high fault rates of the 3D technology call for the incorporation of both prevention and treatment mechanisms in one solution. Previous solutions, as has been included in Figure 4.1, focus on one specific pillar of reliability and provide remedy for that issue. These solutions are usually proposed in isolation of each other, failing to take implications of other aspects into account. Although the mentioned design approach can help to break-down the problem, a narrow design perspective leads to solutions that are difficult to deploy in practice.

This chapter proposes **Reliability by Reconfiguring 3D systems – R2D3**, a holistic, aging-aware reliability engine with fine-grained reconfigurability for parallel streaming systems that can concurrently **detect, diagnose, repair and prevent** failures at runtime. The engine, comprising of a controller, reconfigurable crossbars and detection circuitry, takes advantage of the smaller routing delay over vertical layers in M3D circuits to create a fast-reconfigurable fabric that leverages the availability of identical resources in parallel systems. Figure 4.3 shows high level components of R2D3 incorporated into an in-order

core based parallel 3D system where vertical buses that act as crossbars are inserted between consecutive stages. Our controller creates epochs of execution, at the end of which the detection circuitry can access the inputs and outputs of all layers to verify correctness and diagnose faults. Instead of adding extra redundancy, we salvage *leftovers* which are the functional pipeline stages from faulty cores for fault detection and prevention (orange stages in Fig 4.2). The R2D3 controller manages the reconfigurable fabric to concurrently provide single-cycle replay detection and diagnosis, fault-mitigating repair and aging-aware lifetime management during runtime. To prevent faults, the controller considers temperature variation across the chip and activity factor of each stage to adaptively reconfigure functioning stages to create virtual pipelines. This balances out the aging rate, extending the lifetime of the system. R2D3 can be adopted to M3D parallel architectures that have large number of identical units, such as systolic arrays, mesh-based systems, and many-core or multi-core systems. In this work, we adopt R2D3 for vertically-stacked in-order(OpenSPARC T1) and out-of-order (OoO) pipelines(ARM Cortex A9), as such pipelines closely resemble the basic compute units in modern-day streaming accelerators and massively parallel systems.

We evaluate R2D3, on a physical design of an 8-core, in-order OpenSPARC T1 processor, modified to support fine-grained reconfiguration in 3D, to treat and prevent faults caused by aging, which is analysed by studying the thermal implications. We compare the following 8-core 3D systems: 1) System with R2D3 engine using an adaptive and dynamic reconfiguration policy (**R2D3-Pro**); 2) R2D3 engine using a round-robin dynamic non-adaptive reconfiguration policy (**R2D3-Lite**); 3) System equipped with failure-repairing static reconfiguration policy (**Static**); 4) A 3D-stacked processor with no reconfiguration infrastructure (**NoRecon**). Our evaluation shows that our fault detection technique provides a high coverage of silicon defects (96%). R2D3-Lite and R2D3, respectively, achieve  $1.63\times$  and  $2.16\times$  improvement in lifetime and 52% and 78% increase in throughput over NoRecon, while incurring a marginal 7.4% area and 8.2% frequency overhead in compari-

son to the NoRecon design and negligible overhead in comparison to Static. Furthermore, R2D3 reduces  $V_{th}$  degradation by 53% over a period of 8 years in comparison to NoRecon and Static.

In summary, this chapter makes the following contributions:

1. R2D3 demonstrates how a robust and practical solution should synergistically support all four pillars of reliability: prevention, detection, diagnosis and repair of failures at a low overhead with smart reuse and management of the underlying reconfigurable 3D fabric.
2. Propose R2D3 as a reliability engine that can **detect and diagnosis** faults by re-executing instructions on idle units, incurring a small performance and area overhead. It detects transient and permanent faults and distinguishes them by using a **single-cycle replay**, while localizing the fault to the granularity of a pipeline stage.
3. R2D3 prevents failures by introducing graceful degradation using smart scheduling policies on the same reconfigurable architecture **without any additional overhead** in 3D. It considers the **activity factor** of each stage along with its **temperature variation** by adaptively creating virtual pipelines that balance out the aging rate.

## 4.2 Motivation

The failure rate for electronic systems follow the *bathtub curve* [33] that describes the probability of permanent failure spanning three distinct periods: early failures (fabrication, packaging and shipping), random failures and wearout failures. Although scaling in Si technology has exaggerated all three kinds of failures, years of manufacturing research accompanied with conventional architecture and circuit level reliability solutions have made 2D Si a robust technology [34]. On the contrary, for 3D integration and emerging devices,

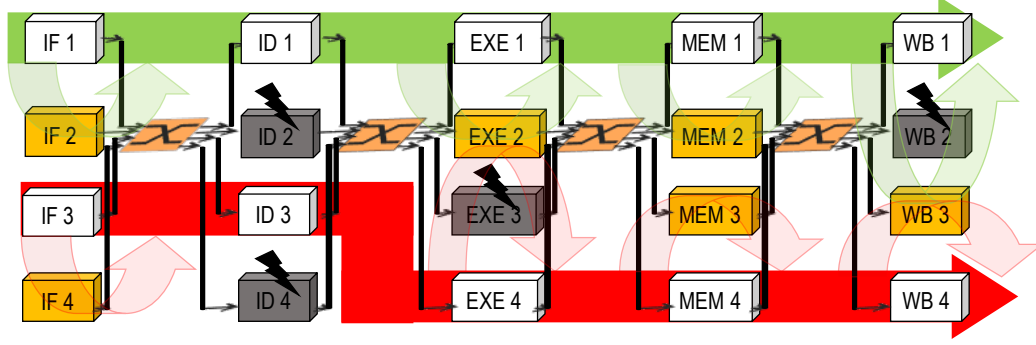


Figure 4.2: Schematic of the R2D3 Engine, where corresponding pipeline stages are stacked vertically and crossbars are inserted between consecutive stages. In this four-fault situation, our solution dynamically reconfigures to connect healthy units as shown with the red and green stripes, providing the compute power of 2 complete cores. Stages in orange are leftovers and still functional and are swap out working stages to distribute wearout and to detect, diagnose and repair faults.

the situation is worse because of the early stages of development and a higher rate of infant and random period errors [7]. Moreover, increased levels of operating temperature, electromigration, current density and thermal variation in 3D technologies speed up the aging failures [35, 36, 37, 38, 39, 40]. Extensive research on both the manufacturing as well as architecture fronts are required to move innovation forward to create large scale chips. However, aggressive manufacturing research usually will not be done unless a product is marketed and products can not be developed profitably because of the challenges in the manufacturing and lifetime, leading to a vicious cycle. Hence, to break this causality dilemma, we present R2D3 as an architectural innovation to improve reliability concerns of the emerging M3D technology.

A naïve reliability solution is to incorporate system-level redundancy, such as dual/triple modular redundancy, which is expensive, inefficient, and only applicable to high-end systems. Prior works leverage the redundant nature of multi-core systems that allow low-cost repair by disabling defective cores using hardware or software mechanisms, which is suitable for standard technologies like 2D silicon technology [92, 102, 46, 103, 104]. However, adopting core-level mechanisms for high fault-rate technologies can cause multiple device failures to discard many cores at once, often times with the majority of the compo-

nents of the faulty core still functional, which could have been easily salvaged. Therefore, these solutions do not scale well with high fault rates. This motivates a rethinking of the architectural fabric from the ground up, with dynamic adaptivity and configurability as primary requirements. To be effective, the fault isolation has to be at a granularity finer than core-level. Although there is a plethora of work to address reliability concerns, no work addresses all the four features of reliability concurrently. Hence, we categorize previous work into fault detection and diagnosis, repair or lifetime management solutions as shown in Table 6.4. The issues with previous solutions can be summarized as follows:

**First**, some solutions lack essential features to be considered practical. For instance, previous reconfigurable architectures such as StageNet [49], Core Cannibal architecture [50], 3DFAR [52] and Viper [47] lack fault detection and diagnosis at a finer granularity which makes them incomplete. **Second**, studies shows that elevated temperatures and longer heat dissipation paths in 3D ICs lead to rapid aging and higher fault rates [35, 36] and this calls for an end-to-end solution that can control aggressive aging and repair the system upon a fault. **Third** is the high-overhead *Frankenstein* issue: while some prior work may seem to incur a lower overhead in a particular category in Table 6.4, they do not provide all the four features of reliability. Even if we ignore compatibility issues and combine multiple solutions together, in a Frankenstein method, the system will incur a high performance penalty and area overhead. For instance, StageNet [49], Core Cannibal architecture [50], and Viper [47] lack fault detection and diagnosis at a fine-granularity which is essential to their repair system. This introduces additional overhead on the top what they propose as fault detection and diagnosis at a fine granularity can be expensive [48, 34]. Moreover, the mentioned solutions miss-out on considering usage and heat leading to non-uniform aging, and hence, lower lifetime and higher faults.

Low yield and high fault rate technologies call for the incorporation of an end-to-end solution and that provides all the four pillars at a low overhead motivating the need for a unified solution at a fine-granularity. To resolve these issues, our solution proposes to utilize

the third dimension, to provide a unified reliability solution with concurrent fault detection, diagnosis, repair and graceful  $V_{th}$  degradation, compared to the limited and specialized approaches mentioned above. With this, R2D3 delivers a substantially reduced-overhead solution, as shown in Table 6.4, when compared to prior work.

#### 4.2.1 Opportunities born from challenges

While future technologies may have low yield and high fault rate, they also provide opportunities like *leftovers* and fast 3D reconfiguration fabric to resolve the challenges.

**Leftovers:** *Leftovers* are salvaged redundant pipeline stages that are functional, but could not be used to form a complete core (orange stages in Figure 4.2). There are two scenarios in which a *leftovers* can be available:

1. Idle functional pipeline stages of faulty cores that cannot form a complete core (orange stages in Figure 4.2).
2. Pipeline stages in other working cores that are powered off temporarily because of a light workload or power and temperature constraints.

Furthermore, the growing gap between the number of cores that can be placed on a chip and those that can be powered-on simultaneously, referred to as the **Many-Core Power Wall** [58], increases the number of available *leftovers*. These *leftovers* can be used to provide the redundancy that is required to support not only treatment but prevention as well. This forms the foundation of our unified reliability solution.

#### 4.2.2 Fast 3D reconfiguration fabric

Monolithic 3D fabric allows to place equivalent resources within short vertical distance from each other, providing a low overhead and fast interconnect network using MIVs. Monolithic 3D technology, by definition, is a 3D integration technology that fabricates two or more layers of devices sequentially, rather than bonding two dies in post-manufacturing

using bumps or TSVs. Cross-layer interconnect switches do not require any buffering, thus greatly simplifying their design and control requirements. Moreover it is possible to use switches to connect pipeline stages both on forward and backward paths. Note that prior reconfigurable solutions like [49], suffer from performance and complexity impacts introduced by buffered switches. However, since propagation delays on vertical MIVs are minimal (approximately two orders of magnitude faster than in conventional 2D layouts) due to the much shorter lengths to be traversed, we can avoid buffering by accommodating a small increase in clock cycle length ( $<8.2\%$ ). This can be extremely important for adopting our solution for more complicated architectures.

### 4.3 Architecture

R2D3 leverages the natural redundancy available in parallel systems to provide robustness against permanent transistor failures. Unlike classic architectures that execute instructions on paths fixed at design time, it relies on inter-stage crossbar switches to form logical pipelines dynamically. Figure 4.3 shows the high level components of R2D3, where corresponding pipeline stages are stacked vertically and vertical buses that function as crossbars are inserted between consecutive stages. By replacing the direct connections at each pipeline stage boundary with interconnect switches, we create a network of resources in which each pipeline stage is connected to all instances of the subsequent stage. At the layer closest to the heat-sink, we insert the reconfiguration controller and detection circuitry which consists of two comparators between subsequent stages, for all layers. The reconfiguration controller can dynamically configure the interconnect to route instructions through functional hardware and detour around failed units, as shown in Figure 4.2.

#### 4.3.1 Interconnect Switch Design

For the interconnect switch design, we adopted a bus-style presented in Chapter II, where vertical links run across the entire height of the design, and each layer uses a set of



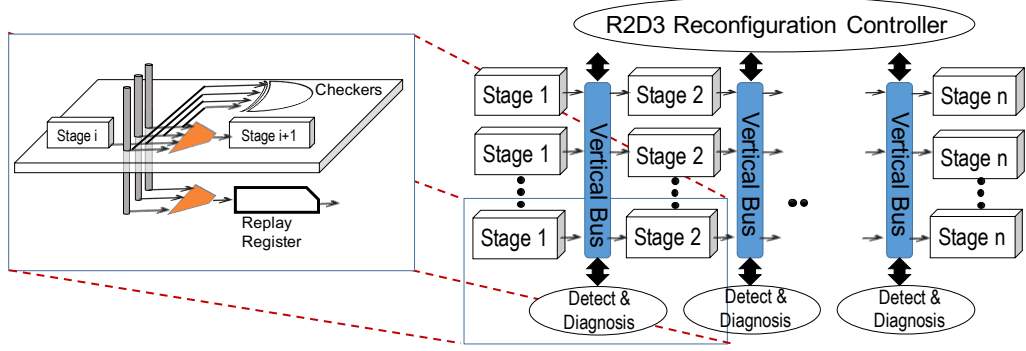


Figure 4.3: Schematic showing high level components of R2D3, where corresponding pipeline stages are stacked vertically and vertical buses that function as crossbars are inserted between consecutive stages. At the end of each epoch, detection circuitry can access the inputs and outputs of all layers to check the output and the R2D3 Reconfiguration Controller manages the components.

multiplexers to select its inputs from the prior stage on the same layer or the prior stage from other layers using the vertical bus lines. The advantage of this solution is that only unidirectional MIVs are required, since signals are switched directly at their destination layer.

To minimize the performance loss from inter-stage communications, we use multiplexer-based full crossbar switches. The MUX-based crossbar has a fixed channel width and, as a result, the delay of transferring an instruction from one stage to the next can occur with a small frequency overhead within the same clock cycle when implemented in 3D [52].

#### 4.3.2 Adaptation for Out-of-Order Architectures

Previous reconfigurable architectures such as StageNet rely on buffered switches with limited bandwidth between consecutive stages. These switches in each pipeline stages boundary complicate the design as different instructions may take different number of clock cycle to reach to the destination and a more complicated mechanisms and control infrastructure for ensuring correct execution. More importantly, these extra buffers between stages introduce performance overhead in terms of IPC as well. Consequently, writers in StageNet paper had to change and redesign the structure of the 5-stage in-order pipeline to

create a 4-stage processor and reduce the number of signals and amount of data between each two stages. The performance overhead and cost of redesigning the whole processor makes the adaptation of these architectures for more complicated architectures such as out-of-order cores or those with more number of stages too expensive. In the case of R2D3, since propagation delays on vertical MIVs are minimal (approximately two orders of magnitude faster than in conventional 2D layouts) due to the much shorter lengths to be traversed, we can avoid buffering by accommodating a small increase in clock cycle length. There are now buffers and added pipeline registers between two consecutive stages and hence, no change needed to modify the architecture of the processor. This is one of the factors that distinguishes R2D3 from previous solution; previous solution in fact are proposing a new architecture with its specific limitations and applications, but this work tries to introduce new framework that can be adopted for any architecture when used in emerging monolithic 3D fabric.

The factor that needs to be considered is the frequency overhead to meet the timing which is dependant on the targeted frequency(how deeply it is pipelined) of the baseline processor and parameters of the 3D technology such as number of vertical layers. The deeper the core is pipelined, the frequency overhead is higher. But keep in mind that the frequency overhead is only felt when the processor is functioning at its maximum frequency which that is not the case many modern processors, usually because of the thermal issues and many core power wall. Moreover, this cost in performance can be mitigated by using more parallel cores at the reduced speed for the demanding workloads which is common in many modern processors.

#### **4.3.3 Cluster Size and Number of Design Layers**

There are contrasting goals in determining the ideal number of layers in a R2D3 design: on one hand the more the layers the more spare units are available, and thus the stronger the robustness of the solution. On the other hand a high number of design layers may

be impractical and may negatively affect the latency required for traversing the vertical dimension of the design to reach a spare unit. In this case, a large many core system would need to be logically divided into smaller R2D3 clusters. Each such cluster would offer full connectivity within itself. Increasing the cluster size will improve reliability as there will be more accessible resources and reconfiguration option in the case of fault. In order to ascertain the right number of layers that can be efficiently stacked together to form a cluster, we can conduct reliability and overhead experiments for different cluster sizes which is investigated in details in experimental results section. We study this trade-off in the experimental evaluation section.

#### 4.4 Fault Detection and Diagnosis

In this section, we introduce our framework to detect the occurrence of faults, distinguish between a transient and permanent fault, and localize them to a single pipeline stage. We take advantage of R2D3’s close vertical distances and fast inter-layer communication network to propose a low-cost and reliable system design approach which provides fine-grained detection and diagnosis of silicon defects at a very low cost and performance overhead. The close proximity of vertical pipeline stages, further helps reduce the overhead associated with the comparison of the output, efficiently finding the failure location. This fabric can be used for runtime validation and failure diagnosis instantly at any layer.

We consider a 3D stacked processor executing an independent instruction stream on each logical pipeline created by the crossbar network and functional stages. Since the executed instruction streams are independent (originating from different threads or tasks), they do not have the same resource utilization (occupancy) profile. This provides us with an opportunity for parallel re-execution of instructions on *leftovers* located in the proximity of the DUT stage.

Our detection and diagnosis scheme utilizes the above concept and does not require any additional redundant hardware. Furthermore, there is near-zero performance penalty

with the concurrent utilization of idle resources. The scheme runs online, *i.e.* it functions in the background during normal application execution and is completely transparent from the software perspective. In this work, we only consider faults in the core data-path and assume a single-fault model; *i.e.*, at each point of time, a single stage becomes faulty.

In brief, our R2D3’s detection and diagnosis procedure achieves the following:

1. Detection: Creates coarse-grained computational epochs that involves the use of *leftovers* to execute identical instructions as those running on the DUT stage.
2. Diagnosis: Distinguishes between a transient fault and a permanent fault, and determines which stage is faulty using a single replay for one clock cycle.
3. In the case of a permanent fault, it isolates the faulty stage, initiates the reconfiguration procedure which constructs logical pipelines based on the latest map of the failed stages and re-executes the task from a checkpoint or the beginning.

## 4.5 Fault Detection

Each epoch takes  $T_{epoch}$  cycles and determines how often a particular stage is tested. At the end of each epoch, we exploit the *leftovers* to verify the functional integrity of each stage of the logical pipelines for  $T_{test}$  cycles by executing identical instructions as those running on the DUT stage. If no fault is detected during the online testing, indicating the underlying hardware is known to be free of silicon defects, the epoch’s computation is ended.

As shown in Figure 4.4, the results are compared using simple inter-stage checkers at the output of the pipeline stages in order to be able to assess the faulty behavior of the involved blocks. Having a high performance inter-layer communication infrastructure, the output and input of each of each pipeline stage can be controlled and monitored at the layer with the crossbar switches. Therefore, if the input of two similar stages in two

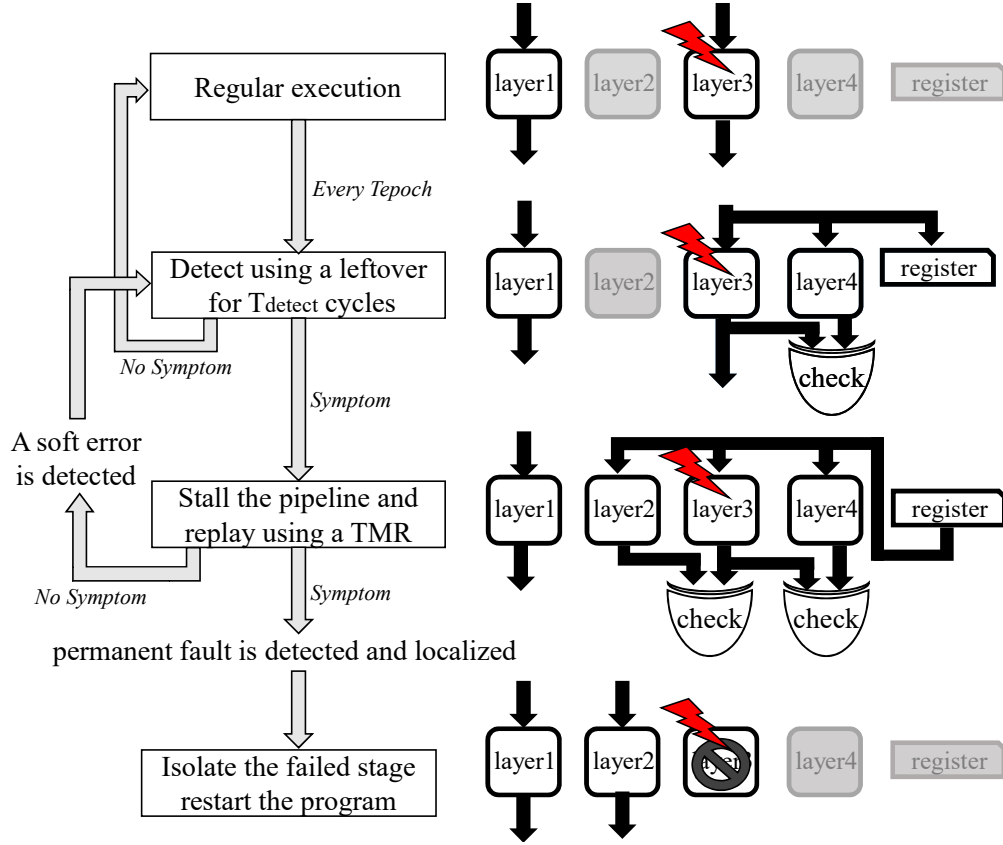


Figure 4.4: R2D3 fault detection, diagnosis and tolerance mechanism in a flowchart (left) and an example on how pipeline stages in a 4-layer design reconfigure (right)

different layers are the same, the output of the two should be identical too. We do not need additional connections as the output and input of each pipeline stage is available within the R2D3 structure in any layer. Note that regardless of the layer that the checkers are placed in, they can access the input and output of all stages through R2D3's vertical buses.

Furthermore, eliminating the performance penalty caused by halting the pipeline for testing provides a unique opportunity to enhance the fault detection. First, it is now possible to evaluate the underlying hardware for a longer period of time (higher  $T_{test}$ ), allowing rigorous testing operation. Second, it is possible to reduce the epoch duration test each stage more frequently (smaller  $T_{epoch}$ ). Hence, the delay for fault detection can be reduced and the performance penalty caused by using the faulty hardware is minimized. On the other hand, using the idle resources for fault detection adds an extra power overhead.

There is a trade-off between test duration/fault coverage ratio and the added power overhead. Our proposed method reduces the area and performance overhead achieving a higher fault coverage rate when compared with previously proposed solutions [48, 34]. As shown in Figure 4.4, we also copy the inputs of the two stages that have been tested into our register for one clock cycle, hence, in the case of a fault we are able to replay them for fault diagnosis.

## 4.6 Fault Diagnosis

After a symptom is detected, the fault needs to be diagnosed (soft-error or permanent) and localized. The diagnosis mechanism distinguishes between transient and permanent hardware faults, and in the case of a permanent fault, identifies the faulty component to initiate system repair. Diagnosis algorithm of R2D3 achieves this with the help of another fault-free stage and the ability to replay an execution using the inputs of the previous cycle, which caused the symptoms, stored in registers as shown in Figure 4.4. R2D3 uses a simple step to distinguish between transient faults and permanent faults and localize it:

It halts the pipeline for one clock cycle and replays the symptom-generating instruction on the two symptom-generating stages and a new known third good stage as it has been shown in Figure 4.4. If the symptom does not recur, a transient fault was detected and the execution continues as regular after having stalled for only one clock cycle. If the symptom recurs on re-execution, the third stage result is compared with the two initial stages and the faulty one is determined and hence, the permanent fault is localized at a pipeline-stage granularity facilitating fine-grained repair and reconfiguration.

Our proposed method has the following desirable properties: (1) No spare units are required - This eliminates a potential single point of system failure. (2) Low hardware overhead - The algorithm uses a lightweight deterministic replay mechanism that does not require capturing memory ordering among different threads. (3) Low performance penalty - Fault detection is concurrent during runtime, and fault diagnosis which is a rare event,

can be achieved with one cycle replay of the previous inputs. (4) Scalability - the algorithm diagnoses a faulty stage at a cost of only one cycle execution replay (the replay required to screen out transients is done concurrently) for any system with  $N \geq 3$  cores.

## 4.7 Repair

By adaptively routing around failed stages, we can salvage working units and repair the system. When a fault occurs, the victim unit (*i.e.* a pipeline stage) is isolated and the reconfiguration procedure is initiated, constructing logical pipelines based on the latest failure map. Since *leftovers* can be on any layer, the executing pipeline may comprise elements from various vertical layers. In Figure 4.2, 4 faults have disabled units on different vertical layers. R2D3 can build 2 complete pipelines dynamically, as shown by the red and green wideband lines, while a core-level solution would have only one functioning core. Once reconfigured, we re-execute the task from a checkpoint or the beginning.

## 4.8 Assumptions and Limitations

In the following, we discuss the our assumptions and limiting factors of our solution.

### 4.8.1 3D vs 2D

We focus on 3D architectures in which elevated temperatures and longer heat dissipation paths due to stacking computational units lead to non-uniform temperature and delay variations across the layers. Although 3D integration has not yet been realized for commercial fabrication, it is one of the most promising emerging solutions to improve transistor density [1]. However, the practicality of its implementation is out of scope of this work. Nonetheless, this framework can be built upon all previous 2D and 3D fault-tolerant reconfigurable solutions, enhancing their performance and lifetime.

### 4.8.2 Addressing Soft Faults

Our approach in its current form is that it can only be used to detect and correct *transient* faults that happen during the test time ( $T_{test}$ ). Zhang et al. [105] characterize soft error vulnerabilities across the stacked layers under 3D integration technology and show that the outer-dies can shield more than 90% of particle strikes for the inner-dies, which leads to an order of magnitude reduction in soft error rates comparing to a similar 2D chip. Hence, for the outer layers more prone to particle strike, our method can assign a longer testing time( $T_{test}$ ) to increase the probability of catching the generated soft error rates. Moreover, critical instructions can be mapped to the more robust inner layers.

### 4.8.3 Caches and System Level Operation

The L0 caches also contain a  $8 \times 8$  crossbar (not shown in Figure 4.2) and have a similar structure to the design in [106]. Hence, caches and TLBs are separated from the stages' logic and considers to be input to the stages which is the case for most modern processors and enables efficient cache pooling for better resource management [106, 21]. This allows seamless effort and no additional overhead from the memory side in the case of re-configuration or running two stages in lockstep for fault detection. Additionally, the cache organization must be set-associative to accommodate both speculative and non-speculative states. Moreover, we assume that faults in local caches are handled with a finer granularity approach like Error-Correcting Codes (ECC).

### 4.8.4 Requirements of Architecture

the use of our approach places a few restrictions on the pipeline and on-chip cache organizations. In particular, the approach of disabling defective functional units requires multiple units of each class. Otherwise, a single defect in a critical non-replicated unit such as central control logic could disable the processor. As such, the stage comparison logic needs to be implemented in TMR logic. The area and frequency overhead may in-



crease based on how tightly the baseline architecture is pipelined. However, considering that modern processors usually use Dynamic Voltage-Frequency Scaling(DVFS) and rarely function at the full frequency, R2D3 frequency overhead really has a small effect over the lifetime of the processor. Simple cores are used in different systems such as Xeon Phi and growing accelerator systems that use in-order cores such as ARM M0 and M4s [106]. These cores will be deployed at much higher scale because of growing demand for accelerators and lack of speculative attack vulnerabilities [107, 108, 109].

#### **4.8.5 Crossbar Reliability**

Moreover, using centralized switches could result in a single point of failure. We leverage a bus-style architecture[52] which is equally distributed between all layers and each layer contains the logic for routing the appropriate instructions for that particular layer where each layer uses a set of multiplexers to select its inputs among one of the vertical bus lines or the prior stage on the same layer and avoiding the single point of failure.

## CHAPTER V

### R2D3 Aging-aware Policy

As discussed earlier, while previous reconfigurable solutions provide fault tolerance, they miss out on the opportunity to leverage this reconfigurability to decelerate aging. Hence, we propose a reconfiguration policy that is aging-aware and provides all treatment methods over the lifetime of the processor. In this subsection we describe these policies.

#### 5.1 Intuition on the Proposed Lifetime Management

In this sections, we demonstrate the effect of temperature profiles on aging in 3D circuits, and provide intuition by an example of uniform  $V_{th}$  degradation due to NBTI via adaptive and dynamic reconfiguration policies.

The key point is that these architectures employ a flexible and fine-grained reconfigurable substrate that provides a great opportunity for creating a complete solution that addresses all aspects of a reliable architecture. Hence, we use the underlying infrastructure in the reconfigurable architectures not only to repair the system, but also to create a fine-grained, low-cost and low-power mechanism for comprehensively detecting transient and permanent errors as well as fault prevention by controlling NBTI-based aging.

Failure to provision for  $V_{th}$  degradation caused by NBTI, as shown in Equation 2.1 and 2.2, causes an increase in delay over time leading to timing failures on critical logic paths. NBTI degradation is frequency independent [69] but is highly impacted by voltage

( $V_{gs}$ ) and temperature ( $T$ ), higher values of which increase the aging rate, as shown in Equation 2.4.

Different heat dissipation paths in 3D ICs lead to non-uniform temperature across its physical layers. Layers that are farther from the heat-sink have higher temperatures and temperature instability, and consequently accelerated  $V_{th}$  degradation. Figure 5.1 shows  $V_{th}$  degradation of different layers in a 4-core processor in 3D over 100 seconds, where each core is placed in a layer. All cores are switched on and off for consecutive periods of 10 seconds making the activity factor of each core equal to 0.5. In this setting, the difference between the HotSpot temperatures of the top and bottom cores is 28 .

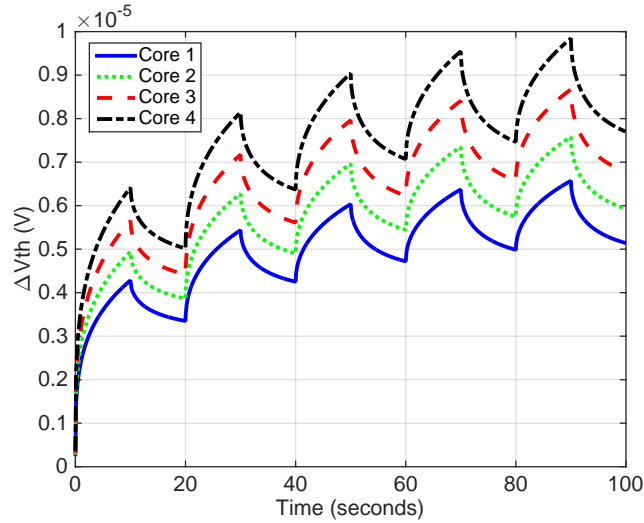


Figure 5.1: Temperature-based  $V_{th}$  degradation of different cores in a 4-core processor in 3D over the duration of 100 seconds with equal on and off time periods (activity factors). All cores are on and then off for periods 10 second and the system does not have any faults. The difference between the hot spot temperature of top and bottom layers is 28 degrees.

For processors with more layers, the difference in temperature between the coolest and hottest cores becomes greater, resulting in a larger gap between their  $V_{th}$  degradation ratios. Increasing the number of cores to 8 elevates the temperature difference of the coolest and hottest cores to 63 which would cause an order of magnitude difference in the NBTI degradation rate of these cores. The increase in supply voltage in response to elevated threshold voltage causes performance degradation and higher power consumption. This

leads to higher temperature and faster aging not only in the corresponding hot cores, but also cores in the adjacent layers. Hence, controlling and balancing the degradation of a multi-core processor in 3D fabric will become more critical in comparison to 2D systems.

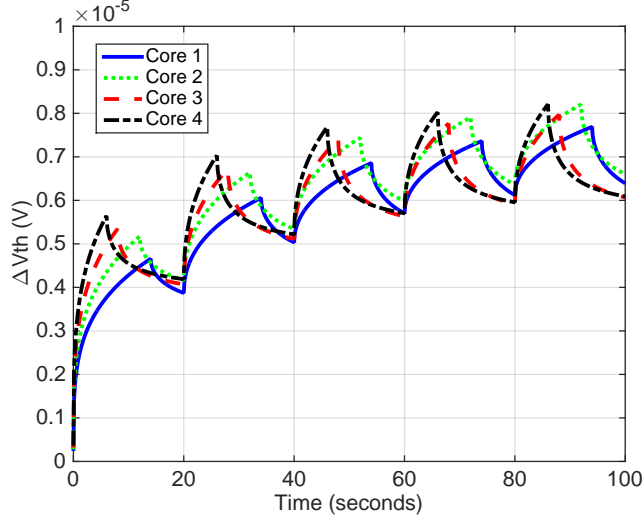


Figure 5.2: Temperature-based  $V_{th}$  degradation of different cores in a 4-core processor in 3D over the duration of 100 seconds with unequal activity factors. The activity factor of each cores is changed so that all cores have the same  $V_{th}$  degradation rate.

Figure 5.2 illustrates  $V_{th}$  degradation of the cores at varying activity factors. For a particular activity factor, due to temperature variations among layers, each core experiences a different  $V_{th}$  degradation. However, the key takeaway and motivation for this work is from the observation that we need to reconfigure pipeline stages to ensure they run at varying activity factors to yield a uniform NBTI-based  $V_{th}$  degradation across all cores on different layers. For instance, to achieve uniform  $\Delta V_{th}=0.008V$  across all cores, the activity factor of each core should be  $\alpha_1 = 0.53$ ,  $\alpha_2 = 0.41$ ,  $\alpha_3 = 0.30$  and  $\alpha_4 = 0.21$ , respectively, where  $\alpha_1$  is the activity factor of the core closest to the heat sink while  $\alpha_4$  is that of the core farthest away in a four core system. By adopting these relatively balanced activity factors, we obtain a more uniform  $V_{th}$  degradation across the cores as illustrated in Figure 5.3. Drawing inspiration from this idea that was demonstrated to balance aging on the core-level, we tune it to the pipeline stage-level for a fine-grained reconfigurable architecture.

In previous solutions, when a new failure appears in a working stage, a static recon-

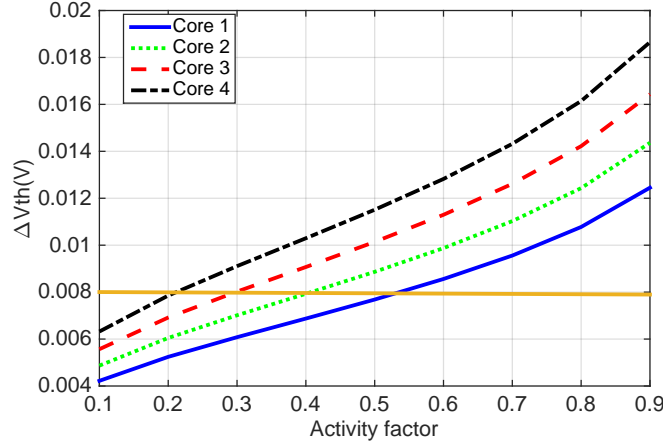


Figure 5.3:  $V_{th}$  degradation of different cores in a 4-core processor in 3D over the duration of 100 seconds by changing their activity factors.

figuration is applied to create logical cores by randomly choosing available functional stages [52]. Initially, the new faulty pipeline is suspended and all processes in execution are swapped out of context by the operating system. Then, the reconfiguration routine computes the number of working pipelines that can be created with the current failure map, and programs the interconnect switches accordingly. The operating system then takes over to reconfigure all the processes in execution, based on the remaining functional pipelines. However, once the reconfiguration is complete and the logical cores have been set up, no further modifications or changes are made unless another fault is detected. Hence, the main difference in comparison to the NoRecon is that reparation is provided using a static reconfiguration policy. However, this scheme uses fixed resources across the 3D stack that are chosen at reconfiguration time, upon a fault, while the *leftover* pipeline stages are shut down. Consequently, the used resources are worn out at a higher rate and experience a larger drop in threshold voltage, reducing the lifetime of the processor considerably.

Alternatively, our solution utilizes *leftover* pipeline stages to provide an opportunity for the resource under stress to partially recover  $V_{th}$ . This enables R2D3 to effectively control and evenly distribute wearout among all working stages. Based on this, we propose two dynamic reconfiguration methods, R2D3-Lite and R2D3 described in the next subsections.

Figure 4.2 demonstrates an example for a specific fault pattern for a 4-core processor, in which two working cores are formed using the functional pipeline stages. In previous reconfigurable architectures the *leftovers* are powered off, but our approach dynamically swaps the stressed stages with *leftovers*. This enables R2D3 to effectively control and evenly distribute wearout among all working stages. Based on this idea, we propose two dynamic reconfiguration methods described in the following subsections.

We optimize the policy to minimize the  $V_{th}$  degradation, and later evaluate its effect on all aging mechanisms.  $V_{th}$  degradation is most serious threat to reliability and cause of failures in systems [41]. However,  $V_{th}$  is partially recovered when the module is no longer under stress (not in use) and optimizing it can reduce the overall aging.

## 5.2 Random Reconfiguration (R2D3-Lite)

To overcome the challenges presented by the previous work, we take advantage of the opportunity that we can replace any working stage with a *leftover* from another layer. Initially, we reassign *leftovers* in a round-robin fashion after a certain reconfiguration time period,  $T_{sched}$ . In this scheme, the policy assigns equal probabilities to choose pipeline stages to create logical cores. For example shown in 4.2, in the case that three execution stages are available and only two of them need to be used based on the application and fault pattern, the probability to use each of them would be 67%. Therefore, the policy switches to the third redundant stage after every  $T_{sched}$  amount of time. Not only does this scheme balance the usage of resources in the processors, but it also gives the units a chance to be unstressed and partially recover their  $V_{th}$  degradation. This policy equalizes the usage of all the pipeline stages, which, however, does not necessarily even out wear-out, as the cores with higher temperatures will degrade faster. This is because it does not consider the varying temperature, a major contributor to NBTI-based aging effects, across the 3D-stack

### 5.3 R2D3

To overcome the shortcomings of a random reconfiguration, R2D3 was specifically designed to address the characteristics of the 3D system. R2D3-Lite considers balancing the usage of each core; however, it does not differentiate between cores in different layers causing non-uniform aging. This is because cores on different layers have different temperatures as they are cooled differently based on their distance to the heat sink. R2D3 assigns an **activity index**,  $A_i$ , to each stage in order to distinguish its temperature and location in the stack. For example, if the activity index of a stage is lower, then it represents that the stage is more prone to hot spots and  $\Delta V_{th}$ . Using the pre-calculated data obtained based on temperature patterns and  $\Delta V_{th}$ , the policy updates these indices during execution and utilizes each pipeline stage based on its activity index while reconfiguring. For instance, to obtain a uniform  $\Delta V_{th} = 0.008V$ , the activity factor of each stage in different cores would be  $\alpha_1 = 0.53$ ,  $\alpha_2 = 0.41$ ,  $\alpha_3 = 0.30$  and  $\alpha_4 = 0.21$ , respectively, where  $\alpha_1$  is the activity factor of the stage in the core closest to the heat sink while  $\alpha_4$  is that of the core farthest away in a four layer system. To balance  $\Delta V_{th}$ , the relative activity factor of the cores needs to follow the predicted values. The policy is modelled to favor stages that are less likely to wear-out and heat-up in the near future by increasing the usage probabilities of cooler stages. Going back to the example in Figure 4.2 where *EXE1*, *EXE2* and *EXE4* are functional, the relative utilization of *EXE2* should be  $\alpha_2/(\alpha_1 + \alpha_2 + \alpha_4)$ . This is the balanced relative utilization for a case that only one running core is needed, and if  $n_{workload}$  is the number of cores that is needed to be run, the activity of *EXE2* also requires to be multiplied by that ratio. This would give us the activity index of *EXE2*,  $A_{EXE2}$ , which is the relative activity that is expected from this stage to run the workload and balance usage as demonstrated in the following equation:

$$A_i = \frac{\alpha_i}{\sum_{j=1}^{n_{live}} \alpha_j} \cdot n_{workload} \quad (5.1)$$

where  $\alpha_i$  is the predicted activity factor for pipeline stage  $i$ ,  $n_{live}$  is the number of pipeline stages of the same type that are functional and available and  $n_{workload}$  is the number of cores required to run the workload and is  $\leq n_{live}$ . The activity indices are calculated every period of calibration window ( $T_{cal}$ ) and each of the available resources are scheduled based on their activity index within  $T_{cal}$  as the following:

$$T_{sched,i} = A_i \cdot T_{cal} \quad (5.2)$$

where  $T_{sched,i}$  is time period over which pipeline stage  $i$  is scheduled to be functionally active. These activity factors can be either determined offline based on the steady state temperature of cores for typical workloads (and therefore implicitly based on the location of cores), or updated at runtime based on the temperature and wear-out ( $\Delta V_{th}$ ) history. In this work, we used predicted activity factors based on the steady state temperature simulations that are described in Section 7.5.

A final advantage is that both of these reconfiguration policies do not entail any additional performance overhead as they do not halt the program execution for swapping the stressed resource with its substitute. To accomplish this, a few cycles before  $T_{sched}$ , we warm up the next stage to be used by duplicating operations and forwarding all the necessary data. Thus, at the time of reconfiguration, the next stage will be substituted in seamlessly and the only action needed would be to shut off the existing working module.



## CHAPTER VI

### R2D3 Evaluation and Results

#### 6.1 Experimental Methodology

This section illustrates the main steps of the proposed reliability evaluation mechanisms to address technology failures and NBTI-based wearout failures. For fault simulation, we use the Synopsys TetraMAX ATPG tool to generate test patterns for the synthesized netlist. In our studies, we explored the stuck-at fault model, which is the industry standard model for test pattern generation. It assumes that a circuit defect behaves as a node stuck at 0 or 1. Our fault model injects permanent transistor failures into any design component and any layer, proportionally to the area of the unit. Once a pipeline unit is hit by a fault, we disable the entire unit and trigger a dynamic reconfiguration. We assume that faults in local caches are handled with ECC. Moreover, if a fault hits an interconnect switch, we disable the unit connected to the output of that switch. If a fault hits the pipeline's control logic, we disable the entire pipeline. We assume that MIVs are implemented reliably as discussed in Section 4.8; we accounted for one spare MIV every 100, based on the recommendation in [93].

A brief overview of our approach is shown in Figure 6.1. We start by considering a given hierarchical description of the design. This description can be provided in any hardware description language such as VHDL or Verilog. In addition, technology parameters are derived based on the technology node in which the design is to be fabricated. In this work, we evaluated Silicon because process parameters and aging models are easier to

acquire. R2D3 can be implemented on top of any emerging device that supports 3D integration, but due to lack of definite wearout and technology models for these technologies, we push the evaluation of R2D3 on these systems to future works.

We synthesized our design on a commercial 45nm SOI process technology with Synopsys Design Compiler, and performed place-and-route of the design using Cadence Innovus. To create the M3D layout of the design, we follow the specifications and design rules recommended by Dae *et al.* in [11] for monolithic 3D integration, and evaluate power and timing through SPICE simulations. The resulting layout represents the block-level floorplan, in which each block is further divided into individual structures or sub-blocks based on the initial structure of the design. This hierarchical design helps us obtain the layout, location, and aspect ratio of each sub-block. The physical design, along with the physical parameters and power consumption estimates generated from Synopsys signoff tools are collected.

The floorplan and power estimates are then fed into HotSpot [110]. HotSpot is an accurate and fast thermal model based on an equivalent circuit of thermal resistance and capacitance that correspond to microarchitectural blocks. We use HotSpot Version 6.0 [110] for thermal modeling. Using the 3D capability available in grid mode, the proposed floorplan of the system is incorporated into the analysis. Also, we use the default characteristics provided by the tool for our die package, as these represent a modern CPU package. The output of the HotSpot simulation is a list with temperatures of each sub-block. The temperature of sub-blocks along with the circuit netlists generated using the Cadence tools are utilized to perform sub-block level SPICE simulations. These simulations provide us with the transistor operating parameters necessary to be plugged into the equations modeling the wear-out mechanisms. It is important to note that the depth of design hierarchy utilized directly impacts the computational runtime, *i.e.* it increases with larger designs.

The core of the proposed methodology relies on a Monte Carlo simulation algorithm implemented and run in MATLAB. Our technique is executed at the sub-block level where

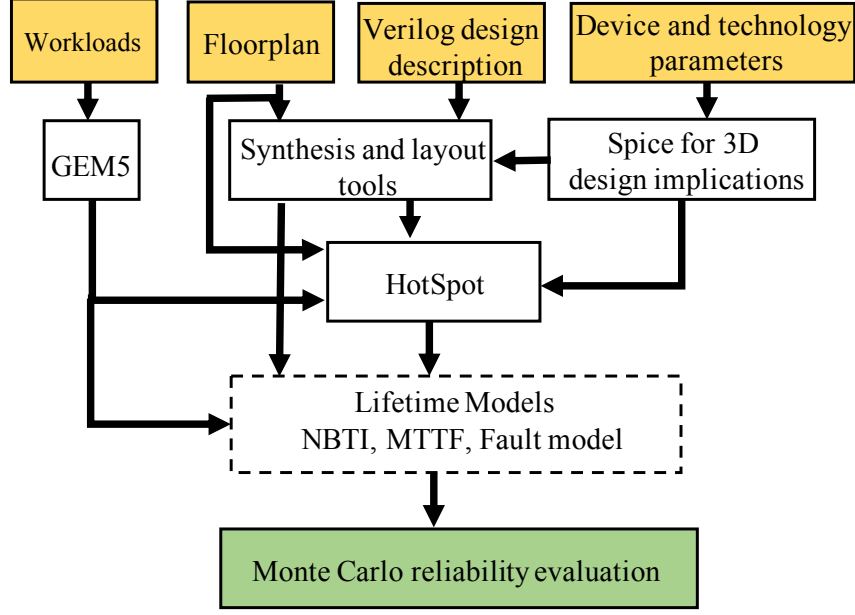


Figure 6.1: Flow chart of the proposed reliability evaluation methodology for NBTI failure mechanism.

the base unit is a transistor. Moreover, at each calibration window, based on the fault pattern and predicted activity factors of the available modules, we calculate the activity index of each core and schedule based on Equation 5.1 and Equation 5.2. As NBTI degradation is independent of frequency, and stress and recovery phases[41, 69], we choose our re-configuration frequency to optimize power and performance overhead, and fault coverage. We set our calibration window for R2D3 equal to 5 ms (5M cycles) which is based on the studies that are presented in Section 6.3.2. During this calibration window, a few cycles before every  $T_{sched}$ , we start the process of handing over from one functional stage to another as described in Section 7.4. Furthermore, to attain statistical confidence, we repeat each experiment on faulty processors 10,000 times, using varying random seeds.

The gem5 simulator [111] was used in this work to simulate the performance of a multicore system with 8 in-order cores. The parameters used for simulating the proposed architecture are shown in Table 6.1. Three popular kernels, general matrix-matrix multiplication (GEMM), general matrix-vector multiplication (GEMV) and fast Fourier transform (FFT), were chosen for performance evaluation. FFT is widely used in communication and visual

processing systems. GEMM and GEMV are ubiquitous kernels in machine learning, scientific workloads and other big data applications. These kernels are often run interchangeably and continually across the lifetime of processors, especially in server machines.

### 6.1.1 Modeling Aging and Mean Time to Failure

This section presents an overview of the major aging and wearout effects. While yield rate is more specific to the process technology and devices, and affect the system early-on, wearout effects impact the entire system significantly across its lifetime. Examples of aging and wearout effects are time-dependent dielectric breakdown (TDDB) [62], negative bias temperature instability (NBTI) [63] and electromigration (EM) [64]. In fact, most wearout mechanisms exhibit an exponential dependence on temperature [65, 66]. The RAMP approach models the mean time to failure (MTTF) of a processor as a function of failure rates of individual structures on-chip associated with high temperature and proposes to calculate the reliability by applying the sum-of-failure-rates (SOFR) model [67]. Vattikonda *et al.* present a divide-and-conquer-based reliability evaluation that employs a Monte Carlo based simulation [68]. As the heart of reliability evaluation in this work is based on Monte Carlo simulations, we adopted the latter. Moreover, as the basis of the developing 3D technology is on silicon transistors, the common aging mechanisms such as NBTI exist and 2D models are widely used for their simulations [35, 36, 39, 40]. All in all, R2D3 is able to decelerate and control non-uniform aging caused by temperature variation through better usage of each resource. Considering that all wearout mechanisms are heavily dependent on temperature and usage, our method can easily be tuned for them. .

### 6.1.2 Fault Model

Our fault model injects permanent transistor failures into any design component and any layer, proportionally to the area of the unit. Once a pipeline unit is hit by a fault, we disable the entire unit and trigger a dynamic reconfiguration via the R2D3 firmware. As discussed

Table 6.1: The gem5 simulation parameters for R2D3.

Module	In-order core parameters	OOO core parameters
<b>Core</b>	SPARC V9 core, 5-stage pipeline @ 1.0 GHz, 3-cycle pipelined integer and floating-point FUs	ARM cortex A-9, 8-stage pipeline @ 1.0 GHz, 4-stage pipelined integer ALU, pipelined floating-point and Load/Store units
<b>L1 D-Cache</b>	8 kB, 4-way set-associative, private cache with 8 MSHRs and 64 B block size with stride prefetcher	32 kB, 4-way set-associative, private cache with 4 MSHRs and 8 B block size with stride prefetcher
<b>L2 D-Cache</b>	64 kB, 4-way set-associative, shared cache with 8 MSHRs and 64 B block size	512 kB, 8-way set-associative, shared cache with 11 MSHRs
<b>I-Cache</b>	4 kB, 4-way set-associative, private cache with 8 MSHRs and 64 B block size	32 kB, 4-way set-associative, private cache with 2 MSHRs
<b>Main Memory</b>	4-channel DDR4-2400 x64 @ 18.8 GB/s per channel	4-channel DDR4-2400 x64 @ 18.8 GB/s per channel

in Section 3.5, faults in local caches are handled with a finer granularity approach. If a fault hits an interconnect switch, we disable the unit connected to the output of that switch. If a fault hits the pipeline’s control logic, we disable the entire pipeline.

This section demonstrates the use of our proposed framework, evaluates our fault detection coverage and compares the lifetime and performance of R2D3 against NoRecon, Static and R2D3-Lite. The underlying system in each case has 8 cores with one physical core in each 3D layer.

## 6.2 Physical Design

We evaluate R2D3 on an OpenSPARC T1 processor which implements the 64-bit SPARC V9 architecture [112]. The OpenSPARC T1 processor contains 8 in-order, five-stage pipelined, single-threaded cores. Each SPARC core has a 16 KB L1 instruction cache, an 8 KB data cache, and fully associative instruction and data translation look-aside buffers (TLB). The 8 cores are connected through a crossbar to an on-chip unified 3 MB L2 cache. For the NoRecon design, we implement a 3D-stacked processor with each core in a new stack and for the Static, R2D3-Lite and R2D3 architectures, we additionally insert vertical bus interconnects. We synthesize and perform place and route of a single core in a commercial 45nm SOI technology using Synopsys Design Compiler and Cadence Innovus tools. To create the 3D layout we followed the specifications and design rules recommended in

[11], and evaluated power and timing through SPICE simulations. The layout obtained is presented in Figure 6.2. Table 6.2 presents the area and power break down based on our physical implementation. Each core has an area of  $0.387 \text{ mm}^2$  and operates at a frequency of 1 GHz. The area occupied by crossbar including MIVs, switching logic and checkers is 7.4% total. Moreover, the frequency decreased by 8.2% because of the delay overhead added by the crossbar units and checkers. We report a power of 250 mW for a single core excluding the power of register files and caches using Synopsys Primetime which shows a 6.1% overhead compared to the NoRecon design. The power derived for each stage is used for thermal simulations as described in Section 6.4.1. There are other smaller components that individually do not contribute significantly to the area and power and are therefore not listed.

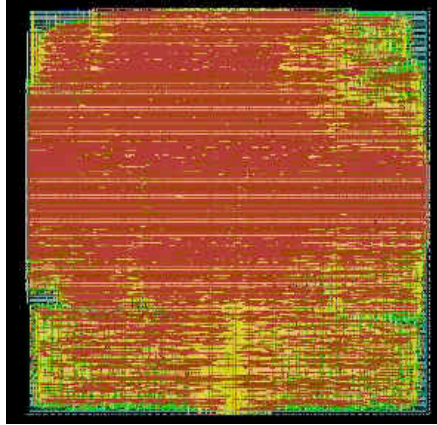


Figure 6.2: Layout of one 3D layer including a complete core (no cache) and all vertical bus switches.

### 6.2.1 Performance in presence of faults

In Figures 6.3 and 6.4, we compare the robustness of R2D3 against the unprotected core(NoRecon) for different number of faults in our two inorder and OOO 8-core 8-layer design that shows how R2D3 provides better performance beyond 1 fault. The performance of the design would be similar when having zero or one fault as the number of working cores that can be created is identical. Two is the minimum number of faults when our

Stage	Total area ( $mm^2$ )	Crossbar Overhead (%)	Checker Overhead (%)	Protected Area (%)	Power (mW)
IFU	0.056	10.3	0.43	88	115
EXU	0.036	12	0.5	95	23
LSU	0.067	18.8	0.74	98	44
TLU	0.040	5	0.22	91	10
FFU	0.014	35.4	1.24	96	3
Total	0.387	7.1	0.31	93	250

Table 6.2: Area and Power for each stage of the 5-stage pipeline.

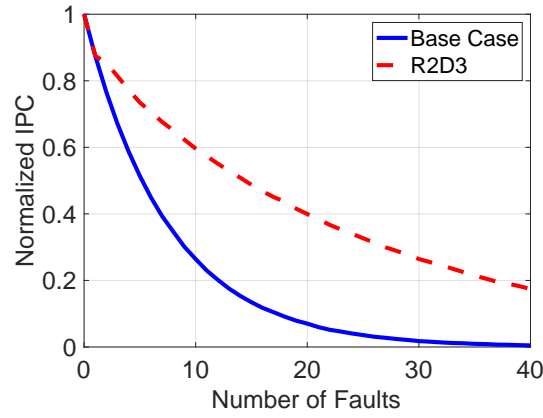


Figure 6.3: Performance of R2D3 with a varying number of concurrent faults for an in-order 8-core 8-layer design

architecture shows improvement as it can use the leftovers to create working pipelines when they don't happen to the same resource or two stages of the same type in two different layers. After 7 faults, R2D3 shows more than  $2\times$  and after 12 fault at least  $3\times$  performance improvement over the NoRecon design. It worth mentioning that the IPC of R2D3 would decrease to less to 10% of the original value after 50 faults. R2D3 has a much more graceful degradation in performance with more faults. R2D3 also does considerably better for the OoO processor (the dotted line in 6.3) compared the in-order case(the dotted line in 6.4). The main reason behind it is the difference in number of stages between the two processors. More the number of stages means a finer granularity which leads to smaller percent of the core being disabled in the case of fault. As R2D3 is taking advantage of

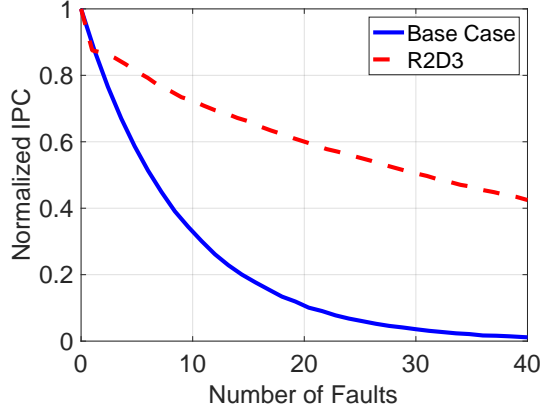


Figure 6.4: Performance of R2D3 with a varying number of concurrent faults in an OOO 8-core 8-layer design

the granularity in the pipeline-stage level, it can provide more flexibility and effect for complicated architecture with higher number of stages.

### 6.3 Fault detection and diagnosis

Based on our fault model that is discussed in the previous section, we assume that effect of the fault can be observed and checked between two identical pipeline stages, which gives us a reasonable fault coverage for targeted hard faults. We performed detailed fault analysis based on the number of test pattern instructions to determine the protected area covered by the fault detection mechanism as reported in Table 6.2.

Figure 6.5 breaks down the types of faults for each unit. The column labeled Total illustrates the additive results for all pipeline stages in the stage level approach, while the Core-Level bar shows the coverage for solutions that target core granularity fault detection. For each case, the bar shows the percentage of detectable faults that are detected, percentage of detectable faults that go undetected when only 10 million ATPG test instructions are run, and the percentage of faults that are undetectable regardless of how many ATPG instructions are run. Figure 6.5 shows the detectable faults (detected plus undetected), which are the target of the detection mechanisms, is 96% for our pipeline-stage-level architec-



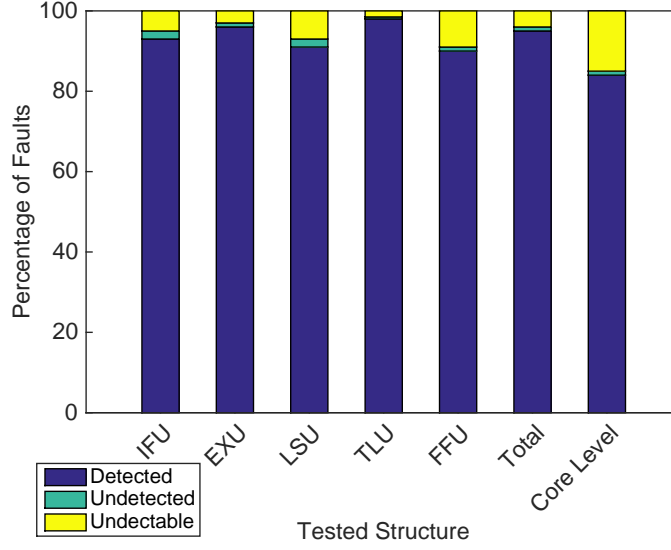


Figure 6.5: Breakdown of different types of faults for each unit. Total illustrates the additive results for all pipeline stages in the stage-level approach and core-level shows the coverage for solutions that have fault detection at a core granularity.

ture (Total) compared to 84% for a core-level mechanism. As our solution gives better observability over signals at the stage level, it is easier to propagate the faults to observable signals.

### 6.3.1 Length of testing period ( $T_{test}$ )

Figure 6.6 gives the breakdown of the percentage of average detectable permanent faults that are detected within a certain number of instructions by running Monte-Carlo fault injection for different applications. For each structure, the bars are divided into several latency stacks from  $<50$  to  $>5K$  instructions. Figure 6.6 shows that, on an average, 96% of detectable permanent faults are detected within 5K clock cycles (the bar labeled as Total). This is considerably higher than the 63% for a core-level architecture. Of course core-level detection is able to achieve higher than 63% coverage rates, but possibly with much longer testing period comparing to R2D3 pipeline-stage level detection. Hence, as 96% is comparatively high among conventional mechanisms (see Table 6.4), we choose 5K cycles as  $T_{test}$ , the length of our fault detection testing period at the end of each epoch. By

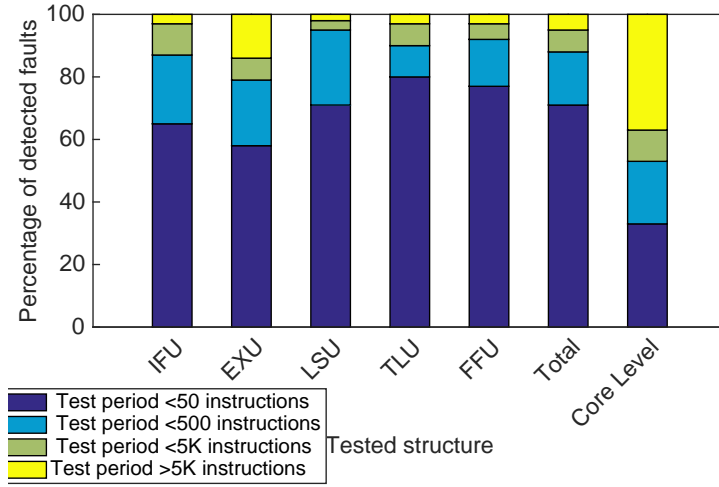


Figure 6.6: Breakdown of average percentage of detected faults of detectable permanent faults by length of testing period  $T_{test}$  for each unit. 96% of the faults are detected within 5K cycles.

increasing  $T_{test}$  the coverage rate will increase gradually, but this will also result in more power overhead as we are running an idle leftover stage in parallel for a longer time. Thus, there is a trade-off between fault coverage and power overhead. Note that as the proposed fault detection policy is concurrent,  $T_{test}$  does not cause any performance overhead.

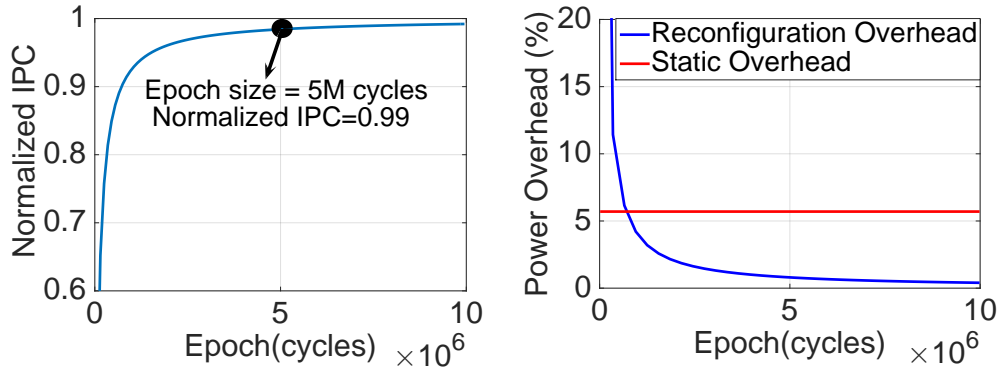


Figure 6.7: Worst case performance degradation (left) and power overhead (right) of re-configuration policy for varying epoch sizes. For the worst case scenario, we assume all 8 cores are always busy and we have to halt one of them to test and run instructions in parallel. For our epoch size (5M cycles) the performance degradation is only 1%.

### 6.3.2 Power Overhead and Epoch Size

Figure 6.7 (right) shows the power overhead of the proposed mechanism for different epoch sizes ( $T_{epoch}$ ). The total number of stacked cores is kept at 8, the size of the  $T_{test}$  is 5K cycles, and 100,000 random fault cases are tested exhaustively for each point on the x-axis. The power penalty is less than 2% when the epoch size is larger than 2M cycles. For an epoch size of 5M cycles, the dynamic power overhead is less than 1%, and thus we select this as our final epoch size. In our design each resource will be checked for permanent faults every 5M cycles or 5ms as our targeted frequency is 1GHz. This determines how  $T_{sched}$  is set, which is the scheduling time for the aging aware mechanism as well. As NBTI degradation is independent of frequency stress and recovery phases [41, 69], we chose our reconfiguration frequency to optimize power and performance overhead and also, fault coverage.

### 6.3.3 Worst Case Overhead

In our proposed fault detection scheme, no performance penalty is incurred, since only idle leftover resources are utilized. But to test the performance overhead of our solution, we design a worst case scenario when all cores are working in parallel continuously. Figure 6.7(left) shows the normalized IPC of a variety of epoch sizes when all cores are always running in parallel. In this case, we need to temporarily suspend one of the cores at the end of each epoch for fault detection of the other ones, which would reduce performance. When the epoch size is 5M cycles, this IPC overhead is 0.8%. The red graph shows the static power overhead, which is almost constant at 5.7%. Overall our mechanism has a 6.5% power overhead when the epoch size is 5M cycles. The mux-based crossbar has a fixed channel width and, as a result, the delay of transferring an instruction from one stage to the next can take place with 8.2% frequency overhead within the same clock cycle of pipeline stage execution based on our implementation.

## 6.4 Lifetime Management

This section evaluates the thermal effects,  $V_{th}$  degradation and the lifetime and performance of R2D3.

### 6.4.1 Thermal Simulation

We use HotSpot Version 6.0 [110] for our thermal modeling. Using the 3D capability available in the grid mode of the tool, the proposed floorplan of the system is incorporated into the analysis. Also, we use the default characteristics provided by the tool for our die package, as these represent a modern CPU package. The thermal sampling interval was 100 ms, which provided sufficient precision. HotSpot was initialized with steady state temperature values.

Layer	1	2	3	4	5	6	7	8
T ( °C)	127	124	121	114	104	93	80	64

Table 6.3: Maximum temperature observed across the layers of an 8-layer design.

The interface material in between the silicon layers is modeled as a homogeneous layer (identified by thermal resistivity and specific heat capacity values) in the thermal model. We computed the thermal impact of the Monolithic Inter-layer Vias (MIVs) connecting the layers by assuming a homogeneous via distribution on the die, and calculated the combined resistivity of the interface material based on the MIV density. Table 6.3 shows the highest HotSpot temperature observed in different layers. The closer the layer is to the heat-sink, the lower its HotSpot temperature is. The temperature of vertically stacked units are correlated with each other, and so, increasing the power in any layer elevates the temperature in the vertically adjacent parts of neighboring layers too.

Although R2D3 is not designed to directly deal with hotspots, it is effective in regulating the temperature by managing the usage of hardware resources based on their temperature profile. It prioritizes utilization of resources that are closer to heat-sink and are

less susceptible to higher temperature (and consequently, faster aging). Figure 6.8 shows the average temperature map of the hottest layer(farthest from the heat-sink) while running different workloads in a loop for Static 3D, R2D3-Lite and R2D3. R2D3 and R2D3-Lite show up to  $33^{\circ}\text{C}$  and  $24^{\circ}\text{C}$  reduction in average temperature over Static.

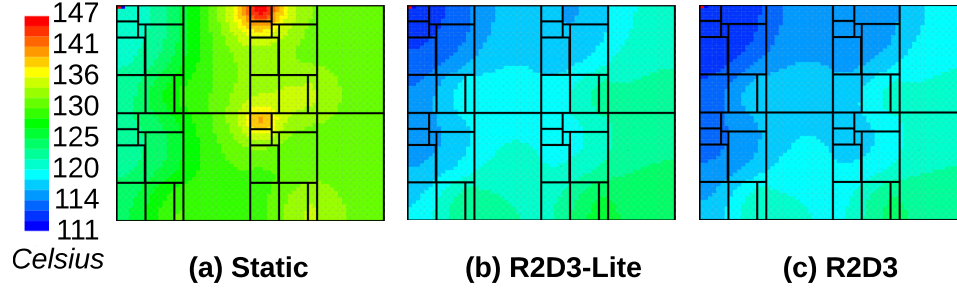


Figure 6.8: Average temperature map of the hottest layer (farthest from the heat-sink) when running different workloads in a loop for (a) Static 3D, (b) R2D3-Lite and (c) R2D3.

#### 6.4.2 NBTI Degradation

Figure 6.9 shows the average  $V_{th}$  degradation map of all layers after 8 years when running different workloads in a loop for NoRecon, Static, R2D3-Lite and R2D3. We evaluate the  $V_{th}$  degradation caused by the NBTI effect across NoRecon, Static, R2D3-Lite and R2D3 policies over a period of 8 years as shown in Figure 6.10.

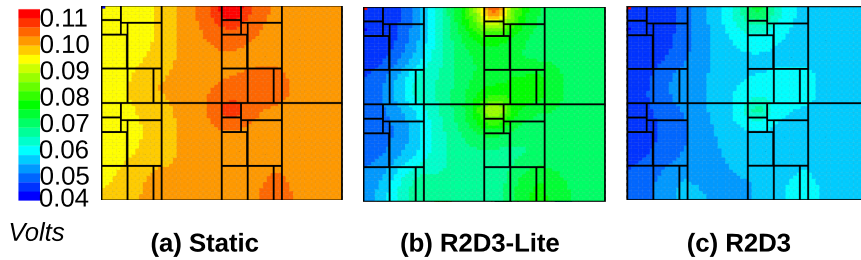


Figure 6.9: Average  $V_{th}$  degradation map of all layers after 8 years when running different workloads in a loop for (a) Static 3D, (b) R2D3-Lite and (c) R2D3.

Although Static has the capability to adaptively repair the system by routing around failed stages to salvage performance, NoRecon and Static degrade identically as neither of them have protections against  $V_{th}$  degradation and will age at the same rate. In contrast,

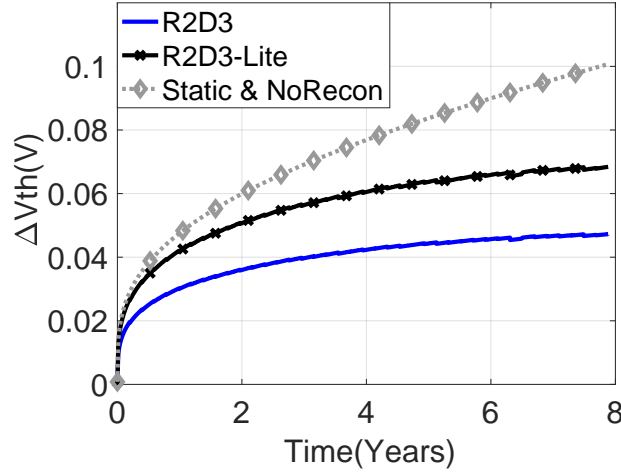


Figure 6.10:  $V_{th}$  degradation for NoRecon, Static, R2D3-Lite and R2D3 over period of 8 years. Degradation for the NoRecon and Static is same, although, Static has much better performance in the presence of faults and failures over the same period of time.

R2D3-Lite and R2D3 are able to reduce the NBTI effect by 31% and 53% over NoRecon and Static. This is because, unlike Static that only uses a chosen set of pipeline stages which are only reconfigured upon a fault, R2D3-Lite and R2D3 manage to balance usage of units across the 3D fabric by swapping stressed stages with *leftovers*. Furthermore, since R2D3 differentiates between stages on different layers, implicitly distinguishing the location of cores, we attain an additional 30% lower  $V_{th}$  degradation over R2D3-Lite.

#### 6.4.2.1 Lifetime and Performance Assessment

Figure 6.12 shows the MTTF comparison between the multicore systems NoRecon, Static, R2D3-Lite and R2D3 over a period of 8 years. The results indicate that assigning relatively lighter workloads on NBTI-stressed devices is beneficial for the recovery of NBTI stress leading to extended device lifespan. On average, R2D3 and R2D3-Lite demonstrate  $2.16\times$  and  $1.63\times$  improvements in MTTF degradation over NoRecon and Static in 8 years. This improvement in lifetime also leads to fault reduction, increased number of working resources and consequently higher performance over an extended period of time. We used our evaluation framework described in Section 7.5 and employed a Monte Carlo

simulation using an area-equivalent fault model to compare the performance of our design. Figure 6.12 illustrates the performance in terms of instructions per cycle (IPC) for the three

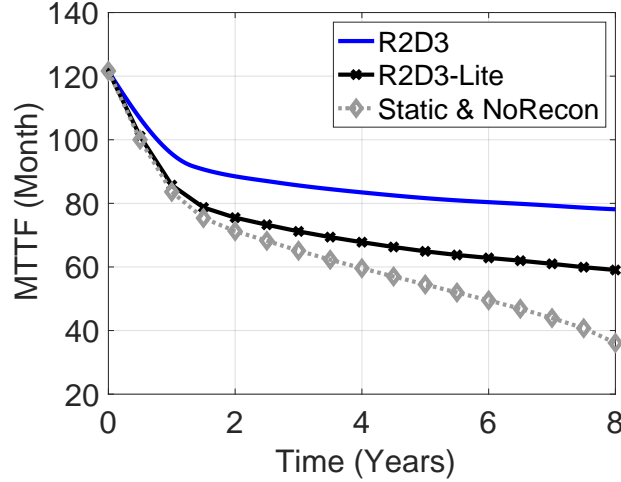


Figure 6.11: Mean Time To Failure for NoRecon 3D, Static, R2D3-Lite and R2D3 over time

workloads running on different configurations over a period of 8 years. Static, R2D3-Lite and R2D3 show improvements over the NoRecon, as they are able to reconfigure in the case of fault occurrences. GEMV achieves a higher performance improvement in comparison to the other two workloads, where R2D3 achieves  $3.76\times$  higher performance over NoRecon at the end of 8 years, compared to  $2.27\times$  and  $1.97\times$  for FFT and GEMM. What distinguishes GEMV from the rest is its highly parallel nature of execution. It exhibits higher utilization, power and temperature and consequently incurs higher aging and failure rates. It also benefits more from the extra cores that R2D3 is able to revive using its reconfigurable fabric.

On average, for the three workloads, R2D3 improves the performance by 78% in comparison to NoRecon over the 8-year period. Moreover, R2D3 and R2D3-Lite improve performance by 21% and 11% over Static. Better reconfiguration and usage of resources based on the locality of 3D fabric leads to decelerated aging rates and mitigated NBTI effects, ultimately improving  $V_{th}$  degradation and performance.

This improvement in lifetime also leads to fault reduction, increased number of working

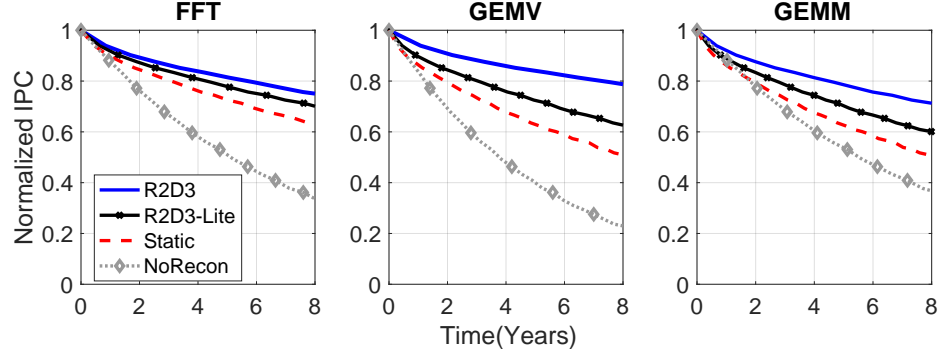


Figure 6.12: Instructions Per Cycle (IPC) of NoRecon 3D, Static, R2D3-Lite and R2D3 running different workloads. In all cases, R2D3 shows much more graceful degradation.

resources and consequently higher performance over an extended period of time. We used our evaluation framework described in Section 7.5 and employed a Monte Carlo simulation using an area equivalent fault model to compare the performance of our design. Figure 6.12 illustrates the performance in terms of Instructions Per Cycle (IPC) for different structures over a period of 8 years. Static, R2D3-Lite and R2D3 show great advantage over the NoRecon design as they are able to reconfigure in the case of each fault occurrence. Moreover, R2D3 and R2D3-Lite improve performance by 21% and 11% on top of Static. Better reconfiguration and usage of resources based on the locality of 3D fabric leads to decelerated aging rates and mitigated NBTI effects, ultimately improving voltage margin and performance.

#### 6.4.2.2 Cluster Size in R2D3

In this section we analyze the impact of increasing the number of layers on area and propagation delay of the interconnect switches. To ascertain the maximum number of layers that can be efficiently stacked together to form a cluster, we evaluated reliability and overhead over varying cluster sizes. Increasing cluster size improves reliability by providing a larger pool of leftovers that can connect and create a working pipeline, but it negatively affects area footprint and interconnect's propagation delay and system's frequency.

Figure 6.13 shows the IPC for different cluster sizes when the total stacked cores are



kept 16. The experiment shows the benefit in reliability by increasing the cluster size, although the gain starts to saturate with the increasing number of cores in a cluster beyond 8 layers. This is so because as there are enough resources in each cluster to create virtual pipelines and the effect of adding more resources to this pool gets more marginal.

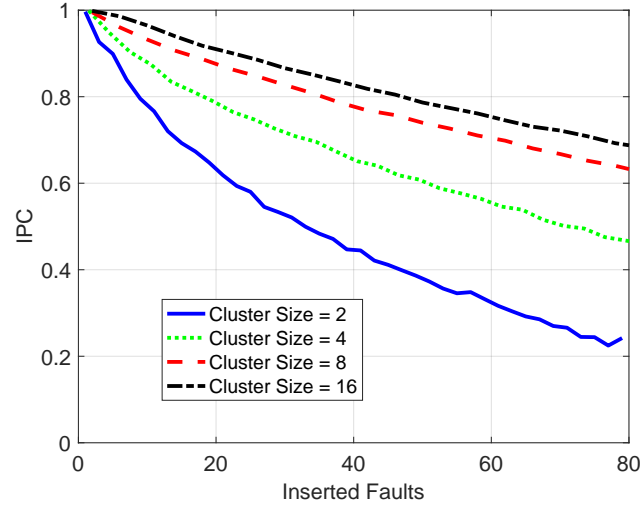


Figure 6.13: Average IPC when cluster size is increasing. The total stacked cores kept at 16 and 10000 random fault cases is tested for each point of x-axis.

The second important factor on determining the cluster size is the wire the delay and area overhead caused by crossbars and vertical connections in cluster. To evaluate the gain from increasing the cluster size, we made an experiment by changing the cluster size for a fixed number of stacked cores at 16. Figure 6.14 shows the area and frequency vs the total cores stacked or the number of slices grouped at a range of cluster sizes. As the number of cores in a cluster increases, the area overhead will increase because of the larger crossbars and more vertical lines required to connect the stages in these cores vertically. The working frequency is decreasing as the number of stacked cores increases because of the delay overhead of the larger crossbars and farther cores in up and bottom of the stacked structure which have to cross more layers to communicate. The results from Figures 6.13 and 6.14 shows that consider cluster size as 8 is a the best option as the gain in reliability starts to fade a way but the overhead in frequency and area starts to dominate the achieved

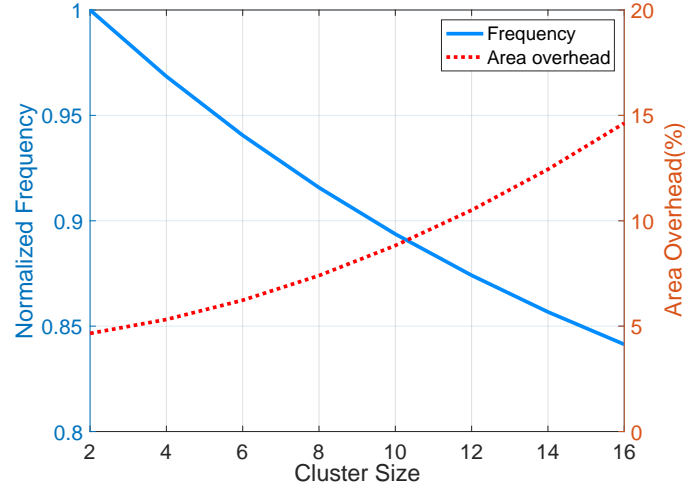


Figure 6.14: Frequency and area of R2D3 for a varying number of 3D layers in a cluster, using the proposed interconnect solution. As the number of cluster size increases, the area overhead increases due to larger crossbars and more MIVs and the frequency will decline due to increased number of layers to cross within a clock cycle, larger crossbars and more MIVs.

gains.

## 6.5 Related Work

Table 6.4 compares the features and performance of R2D3 with related previous works. Generally, we categorize previous solutions into fault detection, fault tolerance and lifetime management techniques. Our technique is the only work that addresses all three aspects and achieves comparable or better metrics than previous solutions while incurring similar amounts of overheads.

### 6.5.1 Fault Detection

Mission critical and high-availability commercial systems, ensure high reliability by using modular redundant configurations, but invest large portions of silicon area and incur performance penalty. Moreover, due to the coarse granularity of redundancy, such systems would still be unable to perform reliably when every processor is subjected to a high fault rate. Some research has focused on developing low-cost fault tolerant techniques for clas-

Reliability Solutions	Granularity	Fault Diagnosis		Fault Tolerance	Lifetime management		Performance cost (%)	Area cost (%)	Power cost (%)
		Detection	Coverage (%)		Technique	Enhancement (%)			
ARGUS [34]	Core	✓	98				3.9	17.0	N.R.
DIVA [92]	Core	✓	N.R.	✓			2.6	N.R.	N.R.
BulletProof [48]	Pipeline stage	✓	89				18.0	5.9	N.R.
ACE [46]	Core	✓	99				5.5	5.8	4.0
mSWAT [102]	Core	✓	95				N.R.	N.R.	N.R.
3DFAR [52]	Pipeline stage			✓			0.0	7.0	N.R.
StageNet [49]	Pipeline stage			✓			33.0	17.0	16.0
Viper [47]	Pipeline stage			✓			24.0	8.0	N.R.
Cobra [113]	Pipeline stage			✓			3.0	N.R.	N.R.
NBTI 3D [114]	Core				✓	Performance: 2.4	9.0	N.R.	N.R.
Bubblewrap [58]	Core				✓	Frequency: 16% Throughput: 30%	N.R.	N.R.	up to 90.0
NBTI Multicore [115]	Core				✓	Failure: 20% MTTF: 30%	6.0	N.R.	N.R.
Facelift [57]	Core				✓	Frequency: 14% Lifetime: 8%	N.R.	N.R.	N.R.
Proactive [116]	Core				✓	Lifetime: 63%	2.0	N.R.	N.R.
Artemis [59]	Core				✓	Performance: 25%	2.0	N.R.	N.R.
R2D3	Pipeline stage	✓	96	✓	✓	Performance: 78% Lifetime: 116%	8.2	7.4	6.1

Table 6.4: Feature Comparison Matrix: Fault detection and diagnosis, repair (tolerance) and prevention (lifetime management) are key features required for a reliable solution which is satisfied only by R2D3 at a low-cost.

\*N.R. stands for not reported

sic pipelined processors that rely on online testing [48], runtime fault detection [34], or defect isolation [49, 47]. In StageNet, removing or reducing interstage communication by breaking loops in the design would be challenging, costly or even not possible in case of more complicated designs. Unfortunately, this solution does not scale to complex designs such as many-core processors. Also, StageNet does not introduce any mechanism for fault detection and prevention using lifetime management.

Bullet-Proof [48] utilizes a microarchitectural checkpointing mechanism that creates coarse-grained epochs of execution, during which distributed on-line built in self-test (BIST) mechanisms validate the integrity of the underlying hardware. BulletProof targets fault detection VLIW designs, but its scalability to multi-core processors is not clear. It also performs poorly in high fault scenarios as it is not a repair mechanism.

Viper [47] proposes a service-oriented execution paradigm. Since it is affected by a number of limitations typical of distributed control architectures, its performance and scalability compares poorly against traditional CMPs and cannot detect faults. In [99], the authors propose to implement the DIVA checker processor [92] in another layer vertically connected to the main processor in 3D structure. They argue that implementing the upper die on an older process that suffers less from soft errors can further increase error tolerance.

Their case study shows that moving to an older technology increases power consumption, but reduces temperature because the power density of the hottest block is lowered. Beside increasing error tolerance, using an older technology in heterogeneous integration can be more cost efficient [117].

Fine-grained architectures can break apart the hardware units of classic hard-wired pipelines, dissolving them into a sea of redundant hardware components. However, these solutions [50, 47, 49, 51, 52] suffer from four key drawbacks. First, they do not address fault detection or diagnosis, which are essential for a holistic treatment solution. They assume that the detection and diagnosis are handled by other mechanisms available in the system. This introduces additional overhead, as fault detection and diagnosis at a fine granularity can be expensive [48, 34]. Second, they do not have a prevention policy that considers usage, heat or aging. Third, the configuration is static until the occurrence of the next fault, leading to non-uniform aging and thermal variations and thus exaggerating runtime failures in 3D systems. Finally, these architectures need dramatic changes to the processor pipeline and control units that makes them complicated and expensive to implement [52].

Utilizing the third dimension can bring ground-breaking opportunities to enhance the performance of integrated systems. In the context of reliability for 3D layouts, R2D3 provides a complete recovery solution with extremely graceful performance degradation, compared to the limited and specialized approaches mentioned above.

### **6.5.2 NBTI and Lifetime Management Techniques**

Traditionally, guardbanding has been used to protect against NBTI. Typically, operating frequency is decreased or supply voltage is increased to account for degradation to prevent timing violations due to aging. Unfortunately, guardbanding incurs a significant performance and power overhead during the entire lifetime, even though NBTI degradation does not fully accumulate until the end. Prior work have proposed several alternatives to mitigate NBTI degradation. These works include adaptive voltage scaling [53], device

oversizing [54], input vector control (IVC) [55] and forward body biasing [56]. These approaches, while trying to accommodate aging, typically affect performance and power. However, they could help to push back the many-core power wall.

Aging is highly dependent on the utilization and operating temperature of a device. System-level techniques take advantage of the application runtime behavior to improve lifetime reliability. Adaptive voltage scaling (AVS) is one such architecture-level technique proposed to mitigate aging in modern processors. Kumar *et. al.* [118] have proposed that instead of a fixed guardband over the entire lifetime of a processor, aging can be reduced by using a lower supply voltage early in the lifetime and increasing the voltage as necessary to counteract the effects of aging. Facelift [57] is a specific application of dynamic voltage scaling (DVS) in which the supply voltage is only adapted once during the lifetime of a processor to switch it from a slow aging mode to a high speed mode. Bubblewrap [58] uses techniques based on Facelift to enhance performance in a multi-core processor. Artemis [59] is an aging-aware application mapping and DVS scheduling framework that considers the PDN-aging of 3D Network-on-Chip (NoC)-based CMPs. These DVS-related works make sense, because the rate of degradation decreases with voltage and due to the front-loaded nature of NBTI, aging effects are reduced during the early phase of the processor’s lifetime.

All of these works only try to mitigate the effects caused by NBTI early on, instead of reducing the aging process over time. The effect is limited, as when the supply voltage increases to counteract aging, the  $V_{th}$  degradation soon converges to that found in the guardbanded case [59]. In contrast, we show that while R2D3 is fault tolerant and does reduce the effects of NBTI, it also helps to reduce the aging process through its smart reconfiguration policy based on the activity and temperature of the design. Another key difference is that our solution manages to control processor degradation at the pipeline-stage-level granularity, while all other solutions consider cores as the smallest subset they can use. Moreover, R2D3’s approach is orthogonal to other solutions such as job allocation

and DVFS, which could also be incorporated into the logical cores underlying R2D3. This means that our solution uses an efficient reconfiguration policy to create logical cores using pipeline stages in different cores, and furthermore, these logical cores can benefit from load balancing and DVFS methods to achieve even better reliability and performance.

### 6.5.3 Comparison to the Frankenstein Solutions

While some solutions may seem to incur a lower overhead in a particular category, they do not provide all the mechanisms R2D3 does and lose out in other categories. If we forcefully combine multiple solutions together, in a Frankenstein method, then we will see major drawbacks in overhead even if we ignore compatibility issues. For example, if we want to create a combined solution with the highest coverage and best lifetime enhancement, including fault repairing, we would combine [116], [52] and [46]. Not only is this impossible since [46] and [52] are incompatible, it would also give a 12.5% performance and 12.8% area overhead without even considering power. R2D3 incurs much lower overheads in comparison to this Frankenstein solution.

## 6.6 Summary

In last three chapters, we proposed R2D3, a solution for the reliability of multi-core processors. R2D3 is the first end-to-end dynamic aging-aware framework built on a fine-grained fault tolerant reconfigurable architecture to detect faults and eliminate/remedy negative bias temperature instability (NBTI) effects at a marginal performance cost.

We propose a low cost framework for fault detection by parallel re-execution of instructions on *leftovers* located in the proximity of the design that achieves 96% coverage rate. Considering temperature variation across a 3D chip and its implications on  $V_{th}$  degradation at different cores in different layers, we show significant slow down of NBTI degradation by swapping in *leftover* pipeline stages to provide an opportunity for the stressed resources to partially recover  $V_{th}$ . Our evaluation, based on performance measurements on a physical

design of an OpenSPARC T1 processor over a period of 8 years, shows that R2D3 reduces  $V_{th}$  degradation by 53% over NoRecon and Static. Furthermore, R2D3 achieves 21% increase in IPC along with a  $2.16\times$  improvement in lifetime over Static, while incurring a marginal 7.4% area and 6.5% power overhead in comparison to the NoRecon 3D design. Our analysis for a worst-case scenario, where there are no idle or *leftover* stages, shows less than a 1% performance overhead.

## CHAPTER VII

# Reliability for CNT Technology

### 7.1 Introduction

With the gradual slowdown of Moore's law, the semiconductor industry has seen the emergence of various new technologies to supplement or replace silicon-based transistors. Although physical scaling of Silicon (Si) is predicted to end with the 5 nm node [75], there is no alternative technology at the moment that has the capability to match the yield and performance of Si for existing designs. Extensive research on both the manufacturing as well as architecture fronts are required to move innovation forward to create large scale chips using these new technologies. However, aggressive manufacturing research is not done unless a product is marketed and products can not be developed profitably because of the high variation in the manufacturing process, leading to a viscous cycle. Hence, to break this causality dilemma, we as architects need to develop design flows for high variation technologies that can overcome the reliability concerns of a new technology and compete with the highly optimized state-of-the-art Si-based designs.

CNTFETs are a promising alternative to Si-based devices due to the exceptional electrical, thermal, mechanical and transport properties, such as high carrier mobility, smaller gate capacitance and better current endurance [5]. Having an order of magnitude better EDP compared to conventional CMOS logic, CNTFET is a promising candidate for building energy-efficient highly integrated digital logic. In this regard, monolithic CNTFET



3D-ICs have been demonstrated by building fully-complementary logic circuits and CNT-FETs on top of silicon CMOS [5]. Recently, CNTFETs have also been adopted to play a key role in the design and development of next-generation of 3D Monolithic System-on-a-Chip technology to create densely integrated logic and memory products [28].

However, the current CNTFET manufacturing process is riddled with imperfections [42]. Chemical synthesis of CNTs does not provide precise control over the locations of individual CNTs on the wafer and consequently, significant variations can exist in the spacing between CNTs. This non-uniformity, which is expressed as *CNT count variation*, leads directly to spatial non-uniformity in the electronic properties of CNTFETs, resulting in increased delay, signal level attenuation and failure. Moreover, a single defective CNTFET can cause faults on the gate level, such as higher leakage, less balanced rise/fall delays or too much driver strength for either pull-up/pull-down path [43]. These problems exist because the technology itself, albeit promising, is still developing from infancy.

As shown in Chapter I, for any disruptive technology to sustain growth and innovation, it usually requires a low-end market with a low barrier of entry to gain initial sources of revenue. With the initial investment, the technology can then improve to enter the mainstream markets and eventually outperform existing technologies. For example, as observed by Christensen in [44], flash memory, a disruptive technology, costed 5-50 $\times$  greater than the hard disk per megabyte of memory and was not as robust for writing. However, flash chips achieve higher performance and reliability at lower power by eliminating moving parts. Flash memory started in small value networks, such as digital cameras, modems and industrial robots. Comparatively, disk drives are too big, fragile and power consuming for these applications. After flash chips succeeded as a niche, the industry began marketing specialized storage systems in portable packages, such as the thumb drive. Today, Solid State Drives (SSDs), made using flash memory, comprise the fastest growing segment of the storage, arriving to this stage because of the initial investments in low-end markets.

Hence, to sustain the development of CNT-based solutions, it is necessary to introduce

architectural and circuit innovations to improve yield, such that we can produce initial designs for the low demand market use. By introducing these solutions, we will help drive the demand for CNT-based products, increasing the number of products that are manufactured. This initial income obtained can be leveraged by the foundry to improve the process technology and expand future lines of products, leading to a stable CNT-based market.

To achieve this goal, we first develop a flexible variation model based on CNT density fluctuations, that allows the designer to mimic the yield obtained for different manufacturing processes. Leveraging the fact that CNTFETs can be used in CMOS logic, we built a 16 nm CNTFET standard cell library and characterized for voltages varying from 0.4 V to 0.7 V to build a design library that can be used to synthesize a logical design. Furthermore, to enable the commercialization of CNTFET-based products, along with the use of the variation model and the design library, we propose the use of a 3D multi-granular reconfigurable architecture, **3DTUBE**, to improve yield and throughput of these high-variation designs. We show that for a failure rate of 10 ppm, employing a pipeline-stage level and module-level reconfigurability helps us achieve a  $2.5\times$  and  $2.0\times$  improvement in performance over a baseline 8-core OpenSPARC T1 processor with no reliability solution, respectively. Furthermore, we also show that by employing these techniques for the current CNT process, we can achieve up to 6x higher throughput and 3.1x lower EDP than that of a 16 nm Silicon-based design at a minimal area cost of 7.4% and frequency overhead of 8.2% over an unprotected multi-core design for the pipeline-stage level and module-level solutions.

## 7.2 Motivation

Deploying technology in a large-scale production requires a high design yield of 3-sigma or above for the product to be profitable. However, due to the infancy of manufacturing processes of emerging technologies, we can not achieve this yield without compromising on size of design.

High-variation observed in the manufacturing process [119] is a significant challenge to be conquered before the benefits of CNTFETs can be reaped. Manufacturing imperfections leading to CNT count variation can affect CNTFET performance and reliability. Despite the strides that have been made in CNTFET manufacturing process over the years, the latest CNTFET process demonstrates a high failure rate of 10 ppm (parts per million) [120]. As seen in Figure 7.1, for a technology with a failure rate of 10 ppm to achieve a yield of 99.73% (3-sigma process), it can only realize a maximum design size of 300 transistors, which is negligible in comparison to the 800 million transistors in the Intel Core i7 processor.

Deriving motivation from the row/column redundancy utilized in SRAMs [121], we observe that by adding redundancy to the design, we can overcome the challenge of low yield and realize larger designs. Adding the redundancy of a second core improves our design size by  $20\times$  as shown in Figure 7.1. Further, adding redundancy at each pipeline-stage and module level improves the size by  $190\times$  and  $255\times$ , respectively, over the no reliability solution design. This allows us to design small reliable CNT processors for the low-end market. However, naively adding redundant cores, pipeline-stages or modules adds expensive area and delay overheads. Prior work has shown successful integration of monolithic 3D circuits using CNTFETs, leveraging the low temperature transfer process of CNTs onto the wafer [5]. We use this ability of CNTFETs to build a 3D architecture that helps minimize the overhead of interconnect delay between redundant logic blocks. Gaining insight from these results, we later demonstrate in Section 7.4, an efficient multi-core 3D architecture that can be reconfigured at multiple granularities to provide an inherent redundancy that improves the yield and performance of CNTFET designs at very low overheads.

### 7.3 Variation Model

Recent fabrication techniques have helped make large scale CNTFET manufacturing processes possible. However, the CNTFET manufacturing is imperfect, due to the pres-

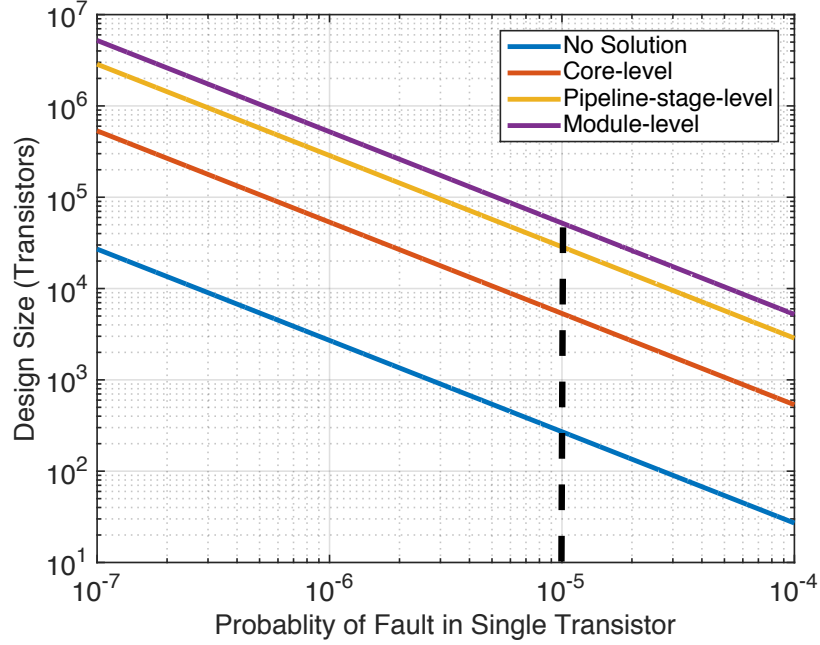


Figure 7.1: Compute size achieved for 99.73% yield if no reliability solution, core-level solution, pipeline-stage level solution or module-level solution is employed for varying fault rates. Dashed line denotes the current process's failure rate of a CNTFET (10 ppm).

ence of metallic carbon nanotubes (m-CNTs), imperfect m-CNT removal processes, chirality drift, CNT doping variations in the source/drain extension regions, diameter variations and density fluctuations due to non-uniform inter-CNT spacing [42]. Based on prior work [122], variation in CNT count caused due to CNT density variations and m-CNT-induced variations is the major contributor of delay degradation, up to 60% in CNTFET circuits at the 16 nm technology node [122]. The lack of precise growth or placement of CNT on a wafer along with the removal of m-CNTs lead to high variation in the CNT density. Furthermore, 3DTUBE can be used to countermeasure any category of CNTFET variation. Therefore for the purposes of this work, we build a flexible variation model based on the fluctuations in the number of CNT and analyze its impact on the yield and performance of CNTFETs.

### 7.3.1 CNT Distribution

To model the CNT distribution observed in CNTFETs, we adopted a gaussian distribution function from [43, 123, 120]. Certain work have also modeled the CNT count distribution as a Weibull distribution [124] or mixed joint distribution [125]. We defer the modeling of these distributions to a future work.

Our model uses the distribution in the inter-CNT spacing, CNT pitch ( $s$ ), to obtain the variation in number of CNTs ( $N$ ). Hence, the distribution of  $N$  can be constructed using:

$$(\mu_N, \sigma_N^2) = (W/\mu_s, W\sigma_s^2/\mu_s^3) \quad (7.1)$$

where  $\mu_N$  is the mean of  $N$ ,  $\sigma_N$  is the standard deviation of  $N$ ,  $W$  is the width of a CNTFET,  $\mu_s$  is the mean of CNT pitch and  $\sigma_s$  is standard deviation of pitch.

Based on the failure rate of 10 ppm (probability of no CNTs using  $W = 2\mu\text{m}$ ,  $N = 9$ ,  $\sigma_N = 2.1$ ) obtained from [120], for a minimum width transistor of 100 nm, we chose a  $\mu_s$  of 20 nm and a  $\sigma_s$  of 10nm for our baseline model and consider a yield failure to occur when  $N \leq 0.25$  CNTs. These parameters can be changed for improvements in the manufacturing process.

#### 7.3.1.1 Effect of CNT Distribution on Yield Failures

Yield failures occur when a transistor has a negligible number of CNTs making it non-functional. To observe the effect of the distribution of CNTs in a device on yield failures, we vary the standard deviation of the pitch from 5 nm to 10 nm (current process technology) as shown in Figure 7.2. This trend shows that a small reduction in the standard deviation of the CNT pitch can reduce yield failures significantly. This analysis is used in Section 7.5 to decide the granularity of our solution. For a high failure rate process, we require redundancy at a fine granularity to create large-scale designs.

### 7.3.1.2 Effect of CNT Distribution on Delay

The distribution of CNT can affect the strength of a transistor, which in turn affects the *delay of a gate* and *performance of a chip*. As shown in Figure 7.2, with increasing standard deviation of pitch, an FO4 (fan-out of 4) inverter's  $3\sigma$  delay is affected *almost linearly*, implying that while the transistor yield rate can have a higher effect on the throughput of a processor, delay variation affects the clock period and timing-based optimization.

### 7.3.2 Variation Suite

When creating huge designs, it is inefficient to obtain the critical paths and find the derate in frequency due to variation in the process technology. Hence, we automate the process by generating a variation suite, which contains the percentage variation in delay for a path of length  $l$ , and average drive strength  $d$ . Note that the creation of this suite is a one-time process for any technology node.

The derate value for a path of length,  $l$ , and average drive strength,  $d$  is calculated as  $3\sigma_{l,d}/\mu_{l,d}$  where,  $\mu_{l,d}$  and  $\sigma_{l,d}$  are the mean and standard deviation of delay obtained from a chain of  $l$  FO4 inverters of strength  $d$ . It has been shown that the delay derate reduces with longer paths and larger gates [126]. Hence, by approximating the original path with the variation seen in inverters, we are slightly pessimistic with our model.

Furthermore, to save design time spent on generating Monte Carlo results for various combinations of depth and strength, we use a statistical model used in static timing tools [127] to estimate the standard deviation for long inverter chains ( $l \geq 5$ ). We estimate the model based on the following equations:

$$\mu_{l,d} = l \cdot \mu_d \quad (7.2)$$

$$\sigma_{l,d} = \sigma_d \cdot \sqrt{l \cdot (1 + 2\rho_d(1 - \frac{1}{l}))} \quad (7.3)$$

where,  $\mu_{l,d}$  and  $\sigma_{l,d}$  are the mean and standard deviation of delay for a path of length  $l$  and strength  $d$ ,  $\mu_d$  and  $\sigma_d$  are the mean and standard deviation of delay of an FO4 inverter with drive strength  $d$  and  $\rho_d$  is the correlation between two adjacent FO4 inverters of strength  $d$ , due to *slew and load capacitance dependence*. On evaluation, we found that our statistical model's variation estimate is within 2% that of the  $3\sigma$  delay variation noticed in the original path for  $l \geq 5$ .

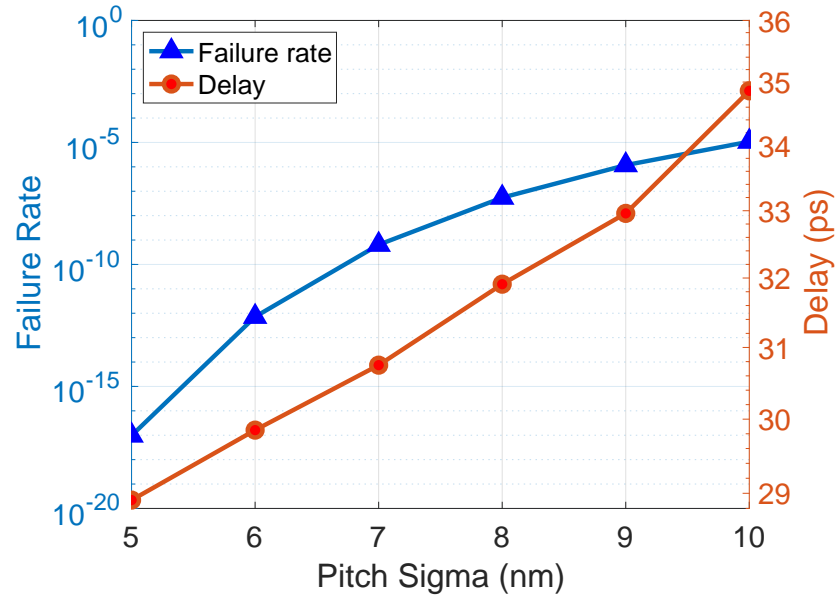


Figure 7.2: Effect of CNT distribution (varying pitch sigma) on FO4 inverter  $3\sigma$  delay and yield failures for  $W = 100$  nm and  $s = 20$  nm

## 7.4 Architecture

Recent reliability solutions propose to break apart the hardware units of classic hard-wired pipelines, dissolving them into a sea of redundant hardware components [49, 52]. Upon fault detection, these designs can reconfigure the hardware by replacing faulty components with new ones. These fine-grained reconfigurable and fault isolating systems can maintain reliability in the presence of faults without sacrificing performance for 2D and 3D systems. Furthermore, leveraging the feasibility of building monolithic CNTFET 3D-

ICs [5], we adopt the solution, 3DFAR to build our solution, 3DTUBE.

3DTUBE is a novel multi-granular, 3D reconfigurable reliability solution for CNT-based processor designs, which leverages the system's natural redundancy to provide robustness against permanent failures. Each manufactured chip, can be configured to route instructions through functioning components and *detour around failed pipeline stages or modules* based on the failure pattern seen in that particular chip, unlike classic architectures that execute instructions on fixed paths.

As shown in Figure 4.2, by replacing the direct connections at the boundary of each pipeline-stage or module with interconnect switches, we create a network of resources in which each component is connected to all instances of the subsequent stage. To minimize the performance loss from inter-stage communications, we use multiplexer-based full crossbar switches because of their non-blocking access to all of their inputs, and small number of input and output ports that are not prohibitively expensive. By adaptively routing around failed stages we can salvage working units to effectively repair the system by creating logical cores across the 3D structure to tolerate variation-based failures.

Based on the manufacturing defects (yield or timing failure) detected in a chip, the victim unit (*i.e.* a pipeline stage or module) is isolated and an identical unit from another layer of the 3D fabric, is used for execution. Hence, the logical core may comprise of elements from various vertical layers. With reference to the example in Figure 4.2, in which 4 defects have disabled units on different vertical layers, our architecture can build two complete cores dynamically (red and green stripes), while a traditional solution (2D or 3D) would have only one.

Figure 7.3 shows two ways; *parallel* or *serial decomposition*; in which the pipeline and the crossbar can be divided to create a modular-level architecture. In parallel decomposition (Figure 7.3.b), the logic and crossbar in each pipeline stage is divided into two parallel parts. In this case no area or delay overhead associated with the crossbars is added compared to the pipeline-level solution (Figure 7.3.a), except new set of control signals



for two different MUXes. In serial decomposition (Figure 7.3.c), the logic and crossbar in each pipeline stage is divided into two sequential parts. The area and delay overhead of the crossbars and vertical connections for  $n$  serial modules would be  $n$  times of the pipeline-stage level solution along with the new set of control signals for  $n$  different MUXes. Hence, we choose parallel decomposition of the pipeline stage as our preferred approach to create the modular-level solution for rest of the chapter. Depending on the defect rate identified, we can adjust the granularity from modular level, pipeline-level or core level obtaining higher yield.

Similar to 3DFAR, 3DTUBE's cross-layer interconnect switches do not require any buffering [52]. Propagation delays on vertical Through Silicon Vias (TSVs) are minimal (more than an order of magnitude faster than in conventional 2D layouts) due to the much shorter lengths to be traversed. Furthermore, to account for manufacturing issues associated with a monolithic 3D-IC, if any of the switching MUX logic fails, the module or pipeline associated with it (component whose input is the MUX's output) in that specific layer would be unusable, but similar stages in other layers, or other stages in that layer would still be functional and can be used to create a logical core. If all MUXes fail, then it would lead to a failure of the entire system. However, as the MUX area is a small percentage of the design compared to the stages, the probability of losing a MUX before the stage it is connected to is negligible. Similarly, reliability is also important when using MIVs (or TSVs), as the failure of a single MIV may cause system failures. Yield and reliability improvements for these MIVs are achieved through a range of redundancy techniques and sparring along with several diagnosis and repair mechanisms [95]. For our design, we account one spare MIV for every 100 MIVs.

## **7.5 Design Flow for High-variation Technology**

This section describes the various design flow steps for a high CNT-density variation technology to be able to produce large scale designs with high yield and performance.

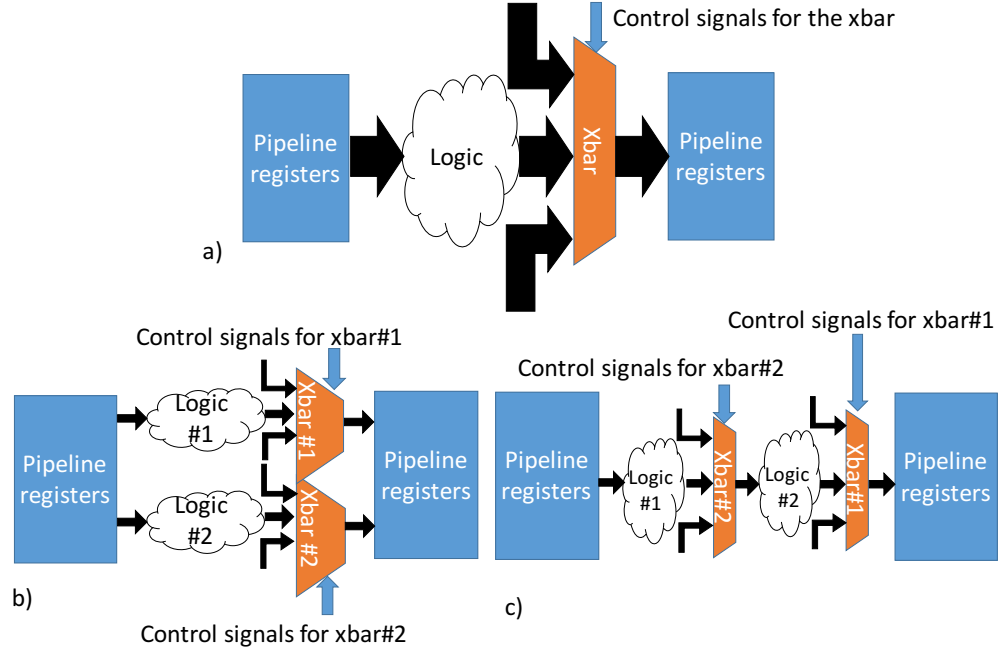


Figure 7.3: a) 3DTUBE pipeline-level structure b) Parallel modular-level architecture c) Serial modular-level architecture

### 7.5.1 Standard Cell Library

The first step in any design flow is to create the basic building blocks, a standard cell library. CNTFETs use carbon nanotubes as the channel medium between the source and the drain, instead of silicon. Leveraging the fact that the behavior of a CNTFET is similar to a Si-FET, *i.e.* we observe a linear region followed by a saturation region in the drain current,  $I_{DS}$ , as a function of increasing gate-source voltage,  $V_{GS}$  [128], we derive CNTFET gates from an already existing 16 nm Si-FinFET cell library to create a 16 nm CNT-based standard cell library. To do so, we match a minimum width 16 nm transistor in the Si-based library to a 100 nm width and 20 nm pitch CNTFET, *i.e.*, a minimum width CNTFET has 5 CNTs.

### 7.5.2 Library Generation

Using the 16 nm CNTFET standard cell library, Stanford's CNTFET Verilog-A model [129] and Cadence's library characterization tool, Liberate, we generate a design library required

to synthesize designs in Synopsys’s Design Compiler for operating voltages varying from 0.4 V to 0.7 V. For a fair comparison of CNT designs generated using this flow, we also generate the design library for the Si-based 16 nm standard cell library.

We demonstrate the results obtained from synthesizing a SPARC-ISA based in-order OpenSPARC T1 CPU core, with no transistor variation, using both the CNT-based and Si-based design libraries for varying voltages in Table 7.1.

Table 7.1: Synthesis results of an OpenSPARC T1 core optimized for performance

<b>Volt. (V)</b>	<b>Delay (ns)</b>		<b>Energy (pJ)</b>		<b>EDP reduction CNT over Si</b>
	CNT	Si	CNT	Si	
0.4	0.65	5.00	7.00	3.97	<b>4.36</b>
0.5	0.50	1.50	9.97	8.65	<b>2.60</b>
0.6	0.40	1.00	14.40	12.81	<b>2.22</b>
0.7	0.35	0.45	20.23	19.87	<b>1.26</b>

### 7.5.3 Design Methodology

We use Stanford’s CNTFET Verilog-A model [129] to create the 16 nm CNTFET standard cell library from a 16 nm Si-based FinFET standard cell library as described in Section 7.5.1. Using this CNTFET cell library and the Verilog-A model, we generate a design library. We also add the variation model to the Verilog-A model as described in Section 7.3. We can use this variation equipped model for both yield and delay analysis of designs. We then generate the variation-suite for path lengths 1 - 5 and drive strength from 1 to 32 using 3-sigma Monte Carlo simulation. For longer path lengths, we use the statistical model to derive an approximate 3-sigma delay variation.

Next, we synthesize an RTL design using the design library to generate a timing report and obtain critical paths to be processed by the variation suite to obtain derated paths (timing of paths adjusted with delay variation). The derated paths and the yield rate of the process are used to derive the granularity at which 3DTUBE has to operate to optimize for design yield and performance. For failure rates greater than  $1.6 \times 10^{-6}$ , we would build a

module-level system, and failure rate less than that, a pipeline-stage-level 3DTUBE.

## 7.6 Evaluation

This section evaluates the usage of 3DTUBE, and compares its performance at multiple granularities against traditional baseline 3D CNTFET and Si designs.

### 7.6.1 Performance and EDP Analysis

We evaluate 3DTUBE on an OpenSPARC T1 processor which implements the 64-bit SPARC V9 architecture [130]. The OpenSPARC T1 processor contains 8 in-order, five-stage pipelined, single-threaded cores. Each SPARC core has a 16 KB L1 instruction cache, an 8 KB data cache, and fully-associative Instruction and Data Translation Look-aside Buffers (TLB). The 8 cores are connected through a crossbar to an on-chip unified 3 MB L2 cache. This processor can achieve an IPC of 1.68 for the SPECJBB 2005 benchmark [130]. The physical design of each core comprises of a total of 80k transistors.

We evaluate the pipeline-level and module-level solutions for operating voltages varying from 0.4 V to 0.7 V by considering two baselines; one, a Si-based design of the OpenSPARC T1 processor evaluated at 1 ppb transistor failure rate and two, a CNT-based design of the processor with no reliability solution. The resulting throughput of our solutions in comparison to the two baselines for differing failure rates can be seen in Figure 7.4. We show the comparison of EDP for two granularities evaluated at 10 ppm over the Si-based baseline at 1 ppb in Table 7.2.

At 0.7 V, for a transistor failure rate of 10 ppm, the module-level design achieves a throughput of 3.1 GOPS (Giga operations per second), pipeline-stage design achieves 2.4 GOPS while the CNT-based baseline achieves 1.3 GOPS throughput for the SPECJBB benchmark, *i.e.*, by employing the module-level technique, we achieve  $2.5\times$  higher throughput over the CNT-based baseline. Although, at 0.4 V, the module-level 3DTUBE design achieves a  $6.0\times$  higher throughput and  $3.1\times$  lower EDP, at 0.7 V, it achieves the same

throughput at 12% higher EDP compared to the silicon-based baseline.

As shown in the Figure 7.4, we can use the module-level optimization for failure rates above 1.6 ppm and the pipeline-stage level optimization to improve throughput for failure rates above 0.1 ppm. For failure rates below 0.1 ppm, the performance of all three designs saturate due to the size of OpenSPARC T1 design. The range of failure rate for the deployment of either levels is highly dependent on the size of the design and is determined during design flow.

Table 7.2: EDP reduction of pipeline-stage and module level solutions of R2D3 for an 8-core OpenSPARC T1 design at 10 ppm transistor failure rate over the Si-based design evaluated at 1 ppb failure rate.

<b>Volt. (V)</b>	<b>EDP Reduction of 3DTUBE over Si</b>	
	<b>Pipeline-level</b>	<b>Module-level</b>
0.4	2.5	3.1
0.5	1.5	1.9
0.6	1.3	1.6
0.7	0.7	0.9

### 7.6.2 Frequency and Area Overhead

To evaluate our architecture, based on detailed measurements done on a physical design, we analyzed the propagation delays for two layouts of a 4-core OpenSPARC T1 processor: first, a 2D layout with the 4 cores in a 2x2 formation with switches placed at the center of the formation and second, a 3D layout with the 4 cores stacked above each other. The worst-case propagation delay for signals going from the output of one stage, through an interconnect switch and to the input of the next stage, for the 2D layout is 20 $\times$  higher when compared to the 3D layout. This vast difference is due to the much shorter distances that must be traversed to reach a corresponding unit in the three-dimensional solution. Based on these findings, our interconnect switch designs and checkers only add an 8.2% frequency overhead.

Furthermore, the pipeline-stage-level framework incurs an overhead of 7.4% in area.

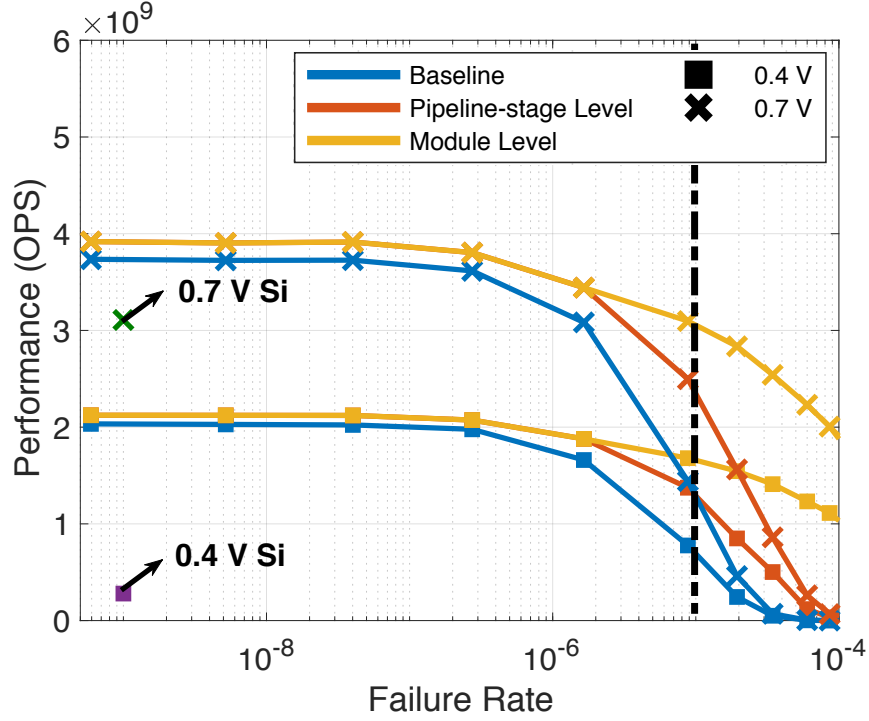


Figure 7.4: Performance of baseline, pipeline-stage level and module level solutions of 3DTUBE for varying transistor failure rate. Dashed line denotes the current process's failure rate of a CNTFET (10 ppm). Si-based design is evaluated at a failure rate of 1 ppb.

The area overhead contains the area of crossbars between stages and TSVs to route the signals across 3D layers. Since propagation delays of the vertical connections are low due to the much shorter lengths to be traversed, we can avoid buffering by accommodating a small increase in clock period for TSVs and MUX-based crossbars in our design. The only real overhead comes with the extra control logic needed as the number of layers increases. For parallel decomposition based modular-level 3DTUBE the only area and delay overhead in addition to the pipeline-stage level solution is generated from new set of control signals which is acceptable for small number of modules in each stage.

## 7.7 Summary

Although breakthrough fabrication techniques to realize carbon nanotube transistors (CNTFETs) have been invented, the process is still at its infancy and has a high failure

rate. We require circuit and architectural reliability solutions to improve the yield and help commercialize CNTFET-based designs, that can pour in investment for fast-paced fabrication improvements. In this work, we propose the use of a 3D reconfigurable architecture, **3DTUBE**, that can provide variation-based failure protection at multiple granularities; module and pipeline-stage levels, to improve yield and throughput of these designs. We show that for a failure rate of 10 ppm, employing a pipeline-stage level and module-level reconfigurability helps us achieve  $2\times$  and  $2.5\times$  improvement in throughput over a CNT-based 8-core OpenSPARC T1 processor with no reliability solution, respectively, at a minimal area cost of 7.4% and frequency overhead of 8.2%. Furthermore, we show that by employing these techniques, we achieve up to  $6.0\times$  higher throughput and up to  $3.1\times$  lower EDP compared to a silicon-based multi-core design evaluated at 1 ppb transistor failure rate, which is  $10,000\times$  lower in comparison to CNTFET's failure rate.

## CHAPTER VIII

### Conclusion and Future Directions

In this work, we leverage recent advances in 3D integration to address these issues and propose a reliable and efficient layout structure and design solution. As device scaling based on Moore’s Law slows down, 3D integration appears to be one of the most promising solutions to continue to increase design density and performance. Although 3D integration presents challenges of its own, including heat dissipation and lack of specialized design tools, they are gaining increasing attention because of their shorter interconnects, higher performance, lower cost, and lower power consumption than corresponding bidimensional fabrics.

Challenges associated with low yield and high fault rates of the 3D technology call for the incorporation of both prevention and treatment mechanisms in one solution. Previous solutions, focus on one specific pillar of reliability and provide remedy for that issue. These solutions are usually proposed in isolation of each other, failing to take implications of other aspects into account. Although the mentioned design approach can help to break-down the problem, a narrow design perspective leads to solutions that are difficult to deploy in practice.

To resolve these issues, our solution proposes to utilize the third dimension, to provide a unified reliability solution with concurrent fault detection, diagnosis, repair and graceful  $V_{th}$  degradation, compared to the limited and specialized approaches mentioned above.



With this, we deliver a substantially reduced-overhead solution when compared to prior work. Our solution proposes to use monolithic 3D fabrics to stack corresponding hardware units from distinct cores above each other, and leverages inter-core redundancy to provide a reliable architecture. We place equivalent resources within short vertical distance from each other, and provide low overhead and fast communication infrastructure using MIVs. By adaptively routing around failed stages we can salvage working units and performance effectively. In developing our solution, we investigated a range of interconnect switch structures, and studied their area overhead and performance.

Although breakthrough fabrication techniques to realize CNTFETs have been invented, the process is still at its infancy and has a high failure rate. We require circuit and architectural reliability solutions to improve the yield and help commercialize CNTFET-based designs, that can pour in investment for fast-paced fabrication improvements. In this work, we propose the use of a 3D reconfigurable architecture, 3DTUBE, that can provide variation-based failure protection at multiple granularities; module and pipeline-stage levels, to improve yield and throughput of these designs.

Extensive research on both the manufacturing as well as architecture fronts are required to move innovation forward to create large scale chips. However, aggressive manufacturing research usually will not be done unless a product is marketed and products can not be developed profitably because of the challenges in the manufacturing and lifetime, leading to a vicious cycle. Hence, to break this causality dilemma, the content of this dissertation is an architectural innovation to improve reliability concerns of the emerging M3D technology.

## 8.1 Future Directions

We recognize several research directions that may extend the contributions of the research proposed in this dissertation.

**Rethinking the conventional architectures:** The degree of freedom that the third dimension provides has a great potential that is yet to be explored. For instance, rather than placing each core in one layer, we can breakdown each core into different layers and place all the same stages in the same layer. This gives the opportunity to place all execute stages to one layer where it could give a great advantage for sharing resources that are not being used. The layer with the most activity can be placed closest to the heat sink to reduce the hot spot. We can also use a superior devices and fabrication technology to reduce power or increase performance for the layer that is the most demanding in terms of computing power.

**Studying the dependencies of this solution on 3D fabrication design rules and parameters:** We used the data and design rules that was provided by studies and papers. Also, there is still a little data from industry and chip foundries on projection of design specifications such as MIV overhead and thickness for 3D technology. It will be interesting to study the dependency of the benefits of our solution on such technology parameters.

**Bringing memory into the picture:** In this dissertation, we assumed that memory is of the chip, but heterogeneous 3D integration offers great opportunities on stacking memory vertically on logic. So it makes sense the investigate the implications of stacking memory on top of the logic levels on reliability and aging of the design.

**Accelerators and GPUs in 3D:** R2D3 can be adopted to parallel architectures that have large number of identical units, such as systolic arrays, mesh-based systems, many-core or multi-core systems. But the for architectures with higher granularity

## **BIBLIOGRAPHY**

## BIBLIOGRAPHY

- [1] “2015 International Technology Roadmap for Semiconductors (ITRS), 2015 Edition”. In: Semiconductor Industry Association, July 2016. URL: <http://www.itrs2.net/itrs-reports.html>.
- [2] Igor Žutić, Jaroslav Fabian, and S Das Sarma. “Spintronics: Fundamentals and applications”. In: *Reviews of modern physics* 76.2 (2004), p. 323.
- [3] Xiaoming Zhu and Joseph M Kahn. “Free-space optical communication through atmospheric turbulence channels”. In: *IEEE Transactions on communications* 50.8 (2002), pp. 1293–1300.
- [4] Geunho Cho et al. “Performance evaluation of CNFET-based logic gates”. In: *I2MTC*. 2009.
- [5] M. M. Shulaker et al. “Monolithic three-dimensional integration of carbon nanotube FETs with silicon CMOS”. In: *2014 Symposium on VLSI Technology (VLSI-Technology): Digest of Technical Papers*. June 2014, pp. 1–2. DOI: 10.1109/VLSIT.2014.6894422.
- [6] Emanuel Knill. “Quantum computing with realistically noisy devices”. In: *Nature* 434.7029 (2005), pp. 39–44.
- [7] M. C. Strus, R. R. Keller, and N. Barbosa. “Electrical reliability and breakdown mechanisms in single-walled carbon nanotubes”. In: *2011 11th IEEE International Conference on Nanotechnology*. Aug. 2011, pp. 715–719.
- [8] H. Wei et al. “Monolithic three-dimensional integration of carbon nanotube FET complementary logic circuits”. In: *2013 IEEE International Electron Devices Meeting*. Dec. 2013, pp. 19.7.1–19.7.4. DOI: 10.1109/IEDM.2013.6724663.
- [9] S. Panth et al. “High-density integration of functional modules using monolithic 3D-IC technology”. In: *Design Automation Conference (ASP-DAC), 2013 18th Asia and South Pacific*. Jan. 2013, pp. 681–686. DOI: 10.1109/ASPDAC.2013.6509679.
- [10] S. Panth et al. “Design challenges and solutions for ultra-high-density monolithic 3D ICs”. In: *Proc. S3S*. 2014.

- [11] K. Yang, D. H. Kim, and S. K. Lim. “Design quality tradeoff studies for 3D ICs built with nano-scale TSVs and devices”. In: *Thirteenth International Symposium on Quality Electronic Design (ISQED)*. Mar. 2012, pp. 740–746. DOI: 10.1109/ISQED.2012.6187574.
- [12] P. Gadfort et al. “A power efficient reconfigurable system-in-stack: 3D integration of accelerators, FPGAs, and DRAM”. In: *2014 27th IEEE International System-on-Chip Conference (SOCC)*. Sept. 2014, pp. 11–16. DOI: 10.1109/SOCC.2014.6948892.
- [13] HT Kung, Bradley McDanel, and Sai Qian Zhang. “Mapping Systolic Arrays onto 3D Circuit Structures: Accelerating Convolutional Neural Network Inference”. In: *2018 IEEE International Workshop on Signal Processing Systems (SiPS)*. IEEE. 2018, pp. 330–336.
- [14] D. Dutoit et al. “A 0.9 pJ/bit, 12.8 GByte/s WideIO memory interface in a 3D-IC NoC-based MPSoC”. In: *2013 Symposium on VLSI Technology*. June 2013, pp. C22–C23.
- [15] Abbas Sheibanyrad, Frdric Ptrot, and Axel Jantsch. *3D Integration for NoC-based SoC Architectures*. 1st. Springer Publishing Company, Incorporated, 2010. ISBN: 1441976175, 9781441976178.
- [16] P. Vivet et al. “A  $4 \times 4 \times 2$  Homogeneous Scalable 3D Network-on-Chip Circuit With 326 MFlit/s 0.66 pJ/b Robust and Fault Tolerant Asynchronous 3D Links”. In: *IEEE Journal of Solid-State Circuits* 52.1 (Jan. 2017), pp. 33–49. ISSN: 0018-9200. DOI: 10.1109/JSSC.2016.2611497.
- [17] O. Hammami, A. M’zah, and K. Hamwi. “Design of 3D-IC for butterfly NOC based 64 PE-multicore: Analysis and design space exploration”. In: *2011 IEEE International 3D Systems Integration Conference (3DIC), 2011 IEEE International*. Jan. 2012, pp. 1–4. DOI: 10.1109/3DIC.2012.6263029.
- [18] Tiansheng Zhang, Jie Meng, and Ayse K. Coskun. “Dynamic Cache Pooling in 3D Multicore Processors”. In: *J. Emerg. Technol. Comput. Syst.* 12.2 (Sept. 2015), 14:1–14:21. ISSN: 1550-4832. DOI: 10.1145/2700247. URL: <http://doi.acm.org/10.1145/2700247>.
- [19] B. K. Joardar, K. Duraisamy, and P. P. Pande. “High performance collective communication-aware 3D Network-on-Chip architectures”. In: *2018 Design, Automation Test in Europe Conference Exhibition (DATE)*. Mar. 2018, pp. 1351–1356. DOI: 10.23919/DATE.2018.8342223.
- [20] S. Das et al. “Monolithic 3D-Enabled High Performance and Energy Efficient Network-on-Chip”. In: *2017 IEEE International Conference on Computer Design (ICCD)*. Nov. 2017, pp. 233–240. DOI: 10.1109/ICCD.2017.43.

- [21] X. Wang et al. "HRC: A 3D NoC Architecture with Genuine Support for Runtime Thermal-Aware Task Management". In: *IEEE Transactions on Computers* 66.10 (Oct. 2017), pp. 1676–1688. ISSN: 0018-9340. DOI: 10.1109/TC.2017.2698456.
- [22] *First 3D Nanotube and RRAM ICs Come Out of Foundry*. <https://spectrum.ieee.org/nanoclast/semiconductors/devices/first-3d-nanotube-and-rram-ics-come-out-of-foundry>.
- [23] D. H. Kim et al. "3D-MAPS: 3D Massively parallel processor with stacked memory". In: *2012 IEEE International Solid-State Circuits Conference*. Feb. 2012, pp. 188–190. DOI: 10.1109/ISSCC.2012.6176969.
- [24] D. Fick et al. "Centip3De: A Cluster-Based NTC Architecture With 64 ARM Cortex-M3 Cores in 3D Stacked 130 nm CMOS". In: *IEEE Journal of Solid-State Circuits* 48.1 (Jan. 2013), pp. 104–117. ISSN: 0018-9200. DOI: 10.1109/JSSC.2012.2222814.
- [25] "Xilinx.Virtex-7 FPGA, available at <http://www.xilinx.com/products/silicon-devices/fpga/virtex-7.html>". In: URL: <http://www.xilinx.com/products/silicon-devices/fpga/virtex-7.html>.
- [26] "MonolithIC 3D ICs, 2015 Edition". In: MonolithIC 3D Inc, Feb. 2013. URL: <http://www.monolithic3d.com/applications.html>.
- [27] M. M. Shulaker et al. "Monolithic 3D integration of logic and memory: Carbon nanotube FETs, resistive RAM, and silicon FETs". In: *2014 IEEE International Electron Devices Meeting*. Dec. 2014, pp. 27.4.1–27.4.4. DOI: 10.1109/IEDM.2014.7047120.
- [28] *SkyWater Begins Work With MIT on Next-Generation Technology Development for DARPA Electronics Resurgence Initiative*. 2018. URL: <https://www.newswire.com/news/skywater-begins-work-with-mit-on-next-generation-technology-20599725>.
- [29] M. M. Sabry Aly et al. "Energy-Efficient Abundant-Data Computing: The N3XT 1,000x". In: *Computer* 48.12 (Dec. 2015), pp. 24–33. ISSN: 0018-9162. DOI: 10.1109/MC.2015.376.
- [30] S. S. Sheu et al. "Fast-Write Resistive RAM (RRAM) for Embedded Applications". In: *IEEE Design Test of Computers* 28.1 (Jan. 2011), pp. 64–71. ISSN: 0740-7475. DOI: 10.1109/MDT.2010.96.
- [31] Yiming Huai. "Spin-transfer torque MRAM (STT-MRAM): Challenges and prospects". In: *AAPPS Bulletin* 18.6 (2008), pp. 33–40.
- [32] M. S. Ebrahimi et al. "Monolithic 3D integration advances and challenges: From technology to system levels". In: *2014 SOI-3D-Subthreshold Microelectronics Tech-*

- nology Unified Conference (S3S)*. Oct. 2014, pp. 1–2. DOI: 10.1109/S3S.2014.7028198.
- [33] Miloš Stanisavljević, Alexandre Schmid, and Yusuf Leblebici. “Reliability, faults, and fault tolerance”. In: *Reliability of Nanoscale Circuits and Systems*. Springer, 2011, pp. 7–18.
  - [34] A. Meixner, M. E. Bauer, and D. Sorin. “Argus: Low-Cost, Comprehensive Error Detection in Simple Cores”. In: *40th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO 2007)*. Dec. 2007, pp. 210–222. DOI: 10.1109/MICRO.2007.18.
  - [35] T. Tanaka. “3D-IC technology and reliability challenges”. In: *2017 17th International Workshop on Junction Technology (IWJT)*. June 2017, pp. 51–53. DOI: 10.23919/IWJT.2017.7966513.
  - [36] Tengfei Jiang et al. “Through-silicon via stress characteristics and reliability impact on 3D integrated circuits”. In: *MRS Bulletin* 40.3 (2015), pp. 248–256.
  - [37] M. I. Khan, A. R. Buzdar, and F. Lin. “Self-heating and reliability issues in Fin-FET and 3D ICs”. In: *2014 12th IEEE International Conference on Solid-State and Integrated Circuit Technology (ICSICT)*. Oct. 2014, pp. 1–3. DOI: 10.1109/ICSICT.2014.7021443.
  - [38] C. S. Premachandran et al. “Reliability challenges for 2.5D/3D integration: An overview”. In: *2018 IEEE International Reliability Physics Symposium (IRPS)*. Mar. 2018, 5B.4-1-5B.4–5. DOI: 10.1109/IRPS.2018.8353609.
  - [39] A. Tsiara et al. “Reliability analysis on low temperature gate stack process steps for 3D sequential integration”. In: *2017 IEEE SOI-3D-Subthreshold Microelectronics Technology Unified Conference (S3S)*. Oct. 2017, pp. 1–3. DOI: 10.1109/S3S.2017.8309219.
  - [40] Y. Li et al. “TSV process-induced MOS reliability degradation”. In: *2018 IEEE International Reliability Physics Symposium (IRPS)*. Mar. 2018, 5B.5-1-5B.5–5. DOI: 10.1109/IRPS.2018.8353610.
  - [41] Hyejeong Hong et al. “Lifetime Reliability Enhancement of Microprocessors: Mitigating the Impact of Negative Bias Temperature Instability”. In: *ACM Comput. Surv.* (Sept. 2015).
  - [42] Carmen Almudever and Antonio Rubio. “Variability & reliability analysis of CN-FET technology: Impact of manufacturing imperfections”. In: *Microelectronics Reliability* (2015), pp. 358–366.
  - [43] J. Zhang et al. “Carbon Nanotube circuits in the presence of carbon nanotube density variations”. In: *DAC*. July 2009, pp. 71–76.

- [44] Clayton Christensen. *The innovator's dilemma: when new technologies cause great firms to fail*. Harvard Business Review Press, 2013.
- [45] S. Gupta et al. "Adaptive online testing for efficient hard fault detection". In: *2009 IEEE International Conference on Computer Design*. Oct. 2009, pp. 343–349. DOI: 10.1109/ICCD.2009.5413132.
- [46] K. Constantinides et al. "A Flexible Software-Based Framework for Online Detection of Hardware Defects". In: *IEEE Transactions on Computers* 58.8 (Aug. 2009), pp. 1063–1079. ISSN: 0018-9340. DOI: 10.1109/TC.2009.52.
- [47] Andrea Pellegrini, Joseph L. Greathouse, and Valeria Bertacco. "Viper: Virtual Pipelines for Enhanced Reliability". In: *Proceedings of the 39th Annual International Symposium on Computer Architecture*. ISCA '12. Portland, Oregon: IEEE Computer Society, 2012, pp. 344–355. ISBN: 978-1-4503-1642-2. URL: <http://dl.acm.org/citation.cfm?id=2337159.2337199>.
- [48] K. Constantinides et al. "BulletProof: a defect-tolerant CMP switch architecture". In: *The Twelfth International Symposium on High-Performance Computer Architecture, 2006*. Feb. 2006, pp. 5–16. DOI: 10.1109/HPCA.2006.1598108.
- [49] S. Gupta et al. "StageNet: A Reconfigurable Fabric for Constructing Dependable CMPs". In: *IEEE Transactions on Computers* 60.1 (Jan. 2011), pp. 5–19. ISSN: 0018-9340. DOI: 10.1109/TC.2010.205.
- [50] B. F. Romanescu and D. J. Sorin. "Core Cannibalization Architecture: Improving lifetime chip performance for multicore processors in the presence of hard faults". In: *2008 International Conference on Parallel Architectures and Compilation Techniques (PACT)*. Oct. 2008, pp. 43–51.
- [51] S. Gupta et al. "Erasing Core Boundaries for Robust and Configurable Performance". In: *2010 43rd Annual IEEE/ACM International Symposium on Microarchitecture*. Dec. 2010, pp. 325–336.
- [52] J. Bagherzadeh and V. Bertacco. "3DFAR: A three-dimensional fabric for reliable multi-core processors". In: *DATE 2017*. 2017.
- [53] L. Zhang and R. P. Dick. "Scheduled voltage scaling for increasing lifetime in the presence of NBTI". In: *2009 ASPDAC*. Jan. 2009.
- [54] S. Roy and D. Z. Pan. "Reliability Aware Gate Sizing Combating NBTI and Oxide Breakdown". In: *2014 27th International Conference on VLSI Design and 2014 13th International Conference on Embedded Systems*. Jan. 2014.
- [55] D. R. Bild, G. E. Bok, and R. P. Dick. "Minimization of NBTI performance degradation using internal node control". In: *2009 DATE*. Apr. 2009.



- [56] H. Mostafa, M. Anis, and M. Elmasry. “NBTI and Process Variations Compensation Circuits Using Adaptive Body Bias”. In: *IEEE Transactions on Semiconductor Manufacturing* (Aug. 2012).
- [57] A. Tiwari and J. Torrellas. “Facelift: Hiding and slowing down aging in multicores”. In: *2008 41st IEEE/ACM Symposium on Microarchitecture*. Nov. 2008.
- [58] U. R. Karpuzcu, B. Greskamp, and J. Torrellas. “The BubbleWrap many-core: Popping cores for sequential acceleration”. In: *2009 42nd MICRO*. Dec. 2009.
- [59] V. Y. Raparti, N. Kapadia, and S. Pasricha. “ARTEMIS: An Aging-Aware Runtime Application Mapping Framework for 3D NoC-Based Chip Multiprocessors”. In: *IEEE Transactions on Multi-Scale Computing Systems* (Apr. 2017).
- [60] Norman P. Jouppi et al. “In-Datcenter Performance Analysis of a Tensor Processing Unit”. In: *44th ISCA*. 2017, pp. 1–12. ISBN: 978-1-4503-4892-8. DOI: 10.1145/3079856.3080246. URL: <http://doi.acm.org/10.1145/3079856.3080246>.
- [61] Yuan Lin et al. “SODA: A Low-power Architecture For Software Radio”. In: *33rd ISCA*. June 2006, pp. 89–101. DOI: 10.1109/ISCA.2006.37.
- [62] R. Moazzami, J. C. Lee, and C. Hu. “Temperature acceleration of time-dependent dielectric breakdown”. In: *IEEE Transactions on Electron Devices* (Nov. 1989).
- [63] T. B. Chan et al. “On the efficacy of NBTI mitigation techniques”. In: *2011 DATE*. Mar. 2011.
- [64] F. M. d’Heurle. “Electromigration and failure in electronics: An introduction”. In: *Proceedings of the IEEE* (Oct. 1971).
- [65] Shantanu Gupta et al. “StageNetSlice: a reconfigurable microarchitecture building block for resilient CMP systems.” In: Jan. 2008, pp. 1–10. DOI: 10.1145/1450095.1450099.
- [66] J. Srinivasan et al. “The impact of technology scaling on lifetime reliability”. In: *International Conference on Dependable Systems and Networks, 2004*. June 2004, pp. 177–186. DOI: 10.1109/DSN.2004.1311888.
- [67] Jayanth Srinivasan et al. “Exploiting structural duplication for lifetime reliability enhancement”. In: *ACM SIGARCH Computer Architecture News*. Vol. 33. 2. IEEE Computer Society. 2005, pp. 520–531.
- [68] JEDEC Specification. “Failure Mechanisms and Models for Semiconductor Devices”. In: *Paper No. JEP 122* (2010), p. 2010.
- [69] R. Vattikonda, Wenping Wang, and Yu Cao. “Modeling and minimization of PMOS NBTI effect for robust nanometer design”. In: *2006 43rd DAC*. July 2006.

- [70] Dieter K. Schroder and Jeff A. Babcock. “Negative bias temperature instability: Road to cross in deep submicron silicon semiconductor manufacturing”. In: *Journal of Applied Physics* 94.1 (2003), pp. 1–18.
- [71] K. C. Wu et al. “Mitigating lifetime underestimation: A system-level approach considering temperature variations and correlations between failure mechanisms”. In: *DATE*. Mar. 2012.
- [72] P Batude et al. “Advances, challenges and opportunities in 3D CMOS sequential integration”. In: *2011 International Electron Devices Meeting*. IEEE. 2011, pp. 7–3.
- [73] M Vinet et al. “3D monolithic integration: Technological challenges and electrical results”. In: *Microelectronic Engineering* 88.4 (2011), pp. 331–335.
- [74] S. Panth et al. “Design challenges and solutions for ultra-high-density monolithic 3D ICs”. In: *2014 SOI-3D-Subthreshold Microelectronics Technology Unified Conference (S3S)*. Oct. 2014, pp. 1–2. DOI: 10.1109/S3S.2014.7028195.
- [75] IEEE. *International Roadmap for Devices and Systems Moore Moore*. Tech. rep. 2018.
- [76] Max M Shulaker et al. “Three-dimensional integration of nanotechnologies for computing and data storage on a single chip”. In: *Nature* 547.7661 (2017), p. 74.
- [77] Saurabh Sinha. “Power benefit study of monolithic 3D IC at the 7nm technology node”. In: *ISLPED 2015* (July 2015), pp. 201–206. DOI: 10.1109/ISLPED.2015.7273514.
- [78] Branimir Radisavljevic et al. “Single-layer MoS<sub>2</sub> transistors”. In: *Nature nanotechnology* 6.3 (2011), p. 147.
- [79] Omid Habibpour et al. “Wafer scale millimeter-wave integrated circuits based on epitaxial graphene in high data rate communication”. In: *Scientific reports* 7 (2017), p. 41828.
- [80] Mohamed M Sabry Aly et al. “Energy-efficient abundant-data computing: The N3XT 1,000 x”. In: *Computer* 48.12 (2015), pp. 24–33.
- [81] Phaeton Avouris, Zhihong Chen, and Vasili Perebeinos. “Carbon-based electronics”. In: *Nanoscience And Technology: A Collection of Reviews from Nature Journals*. World Scientific, 2010, pp. 174–184.
- [82] H-S Philip Wong and Deji Akinwande. *Carbon nanotube and graphene device physics*. Cambridge University Press, 2011.
- [83] M Radosavljević et al. “Drain voltage scaling in carbon nanotube transistors”. In: *Applied Physics Letters* 83.12 (2003), pp. 2435–2437.

- [84] H. Wei et al. “Monolithic three-dimensional integration of carbon nanotube FET complementary logic circuits”. In: *2013 IEEE International Electron Devices Meeting*. Dec. 2013, pp. 19.7.1–19.7.4. DOI: 10.1109/IEDM.2013.6724663.
- [85] M. M. Shulaker et al. “Monolithic three-dimensional integration of carbon nanotube FETs with silicon CMOS”. In: *VLSI Technology (VLSI-Technology): Digest of Technical Papers, 2014 Symposium on*. June 2014, pp. 1–2. DOI: 10.1109/VLSIT.2014.6894422.
- [86] J. Deng et al. “Carbon Nanotube Transistor Circuits: Circuit-Level Performance Benchmarking and Design Options for Living with Imperfections”. In: *2007 IEEE International Solid-State Circuits Conference. Digest of Technical Papers*. Feb. 2007, pp. 70–588. DOI: 10.1109/ISSCC.2007.373592.
- [87] Aaron D Franklin et al. “Sub-10 nm carbon nanotube transistor”. In: *Nano letters* 12.2 (2012), pp. 758–762.
- [88] M. M. Shulaker et al. “Monolithic three-dimensional integration of carbon nanotube FETs with silicon CMOS”. In: *VLSI Technology (VLSI-Technology): Digest of Technical Papers, 2014 Symposium on*. June 2014, pp. 1–2. DOI: 10.1109/VLSIT.2014.6894422.
- [89] Wangyuan Zhang and Tao Li. “Microarchitecture soft error vulnerability characterization and mitigation under 3D integration technology”. In: *2008 41st IEEE/ACM International Symposium on Microarchitecture*. Nov. 2008, pp. 435–446. DOI: 10.1109/MICRO.2008.4771811.
- [90] Sparsh Mittal and Jeffrey S Vetter. “A survey of techniques for modeling and improving reliability of computing systems”. In: *IEEE Transactions on Parallel and Distributed Systems* 27.4 (2016), pp. 1226–1238.
- [91] I. Koren and S. Y. H. Su. “Reliability Analysis of N-Modular Redundancy Systems with Intermittent and Permanent Faults”. In: *IEEE Transactions on Computers* C-28.7 (July 1979), pp. 514–520. ISSN: 0018-9340. DOI: 10.1109/TC.1979.1675397.
- [92] Todd M. Austin. “DIVA: A Reliable Substrate for Deep Submicron Microarchitecture Design”. In: *Proceedings of the 32Nd Annual ACM/IEEE International Symposium on Microarchitecture*. MICRO 32. Haifa, Israel: IEEE Computer Society, 1999, pp. 196–207. ISBN: 0-7695-0437-X. URL: <http://dl.acm.org/citation.cfm?id=320080.320111>.
- [93] L. Jiang et al. “On effective and efficient in-field TSV repair for stacked 3D ICs”. In: *Design Automation Conference (DAC), 2013 50th ACM/EDAC/IEEE*. May 2013, pp. 1–6. DOI: 10.1145/2463209.2488824.
- [94] Robert Swarz and Daniel Siewiorek. *Reliable computer systems: design and evaluation*. 3rd. A K Peters/CRC Press, 1998.

- [95] S. Wang, M. B. Tahoori, and K. Chakrabarty. “Defect clustering-aware spare-TSV allocation for 3D ICs”. In: *Computer-Aided Design (ICCAD), 2015 IEEE/ACM International Conference on*. Nov. 2015, pp. 307–314. DOI: 10.1109/ICCAD.2015.7372585.
- [96] C. C. Yang, C. W. Chou, and J. F. Li. “A TSV Repair Scheme Using Enhanced Test Access Architecture for 3-D ICs”. In: *2013 22nd Asian Test Symposium*. Nov. 2013, pp. 7–12. DOI: 10.1109/ATS.2013.12.
- [97] S. Hari et al. “mSWAT: Low-cost hardware fault detection and diagnosis for multi-core systems”. In: *Proc. MICRO*. 2009.
- [98] S. Safiruddin et al. “Zero-performance-overhead online fault detection and diagnosis in 3D stacked integrated circuits”. In: *2012 IEEE/ACM International Symposium on Nanoscale Architectures (NANOARCH)*. July 2012, pp. 123–130. DOI: 10.1145/2765491.2765514.
- [99] Niti Madan and Rajeev Balasubramonian. “Leveraging 3D Technology for Improved Reliability”. In: *Proceedings of the 40th Annual IEEE/ACM International Symposium on Microarchitecture*. MICRO 40. Washington, DC, USA: IEEE Computer Society, 2007, pp. 223–235. ISBN: 0-7695-3047-8. DOI: 10.1109/MICRO.2007.22. URL: <http://dx.doi.org/10.1109/MICRO.2007.22>.
- [100] B. Ghavami and M. Raji. “Failure Characterization of Carbon Nanotube FETs Under Process Variations: Technology Scaling Issues”. In: *IEEE Transactions on Device and Materials Reliability* 16.2 (June 2016), pp. 164–171. ISSN: 1530-4388. DOI: 10.1109/TDMR.2016.2543659.
- [101] Ahmad Islam. “Variability and Reliability of Single-Walled Carbon Nanotube Field Effect Transistors”. In: *Electronics* 2 (Dec. 2013), pp. 332–367. DOI: 10.3390/electronics2040332.
- [102] S. K. S. Hari et al. “mSWAT: Low-cost hardware fault detection and diagnosis for multicore systems”. In: *2009 42nd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*. Dec. 2009, pp. 122–132.
- [103] S. Nomura et al. “Sampling + DMR: Practical and low-overhead permanent fault detection”. In: *2011 38th Annual International Symposium on Computer Architecture (ISCA)*. June 2011, pp. 201–212. DOI: 10.1145/2000064.2000089.
- [104] Jared C Smolens et al. “Detecting emerging wearout faults”. In: *Proc. of Workshop on SELSE*. 2007.
- [105] Wangyuan Zhang and Tao Li. “Microarchitecture soft error vulnerability characterization and mitigation under 3D integration technology”. In: *2008 41st IEEE/ACM International Symposium on Microarchitecture*. Nov. 2008, pp. 435–446. DOI: 10.1109/MICRO.2008.4771811.

- [106] S. Pal et al. “OuterSPACE: An Outer Product Based Sparse Matrix Multiplication Accelerator”. In: *2018 IEEE International Symposium on High Performance Computer Architecture (HPCA)*. Feb. 2018, pp. 724–736. DOI: 10.1109/HPCA.2018.00067.
- [107] Paul Kocher et al. “Spectre Attacks: Exploiting Speculative Execution”. In: *CoRR* abs/1801.01203 (2018). arXiv: 1801.01203. URL: <http://arxiv.org/abs/1801.01203>.
- [108] Moritz Lipp et al. “Meltdown”. In: *CoRR* abs/1801.01207 (2018). arXiv: 1801.01207. URL: <http://arxiv.org/abs/1801.01207>.
- [109] Jo Van Bulck et al. “Foreshadow: Extracting the Keys to the Intel SGX Kingdom with Transient Out-of-Order Execution”. In: *27th USENIX Security Symposium (USENIX Security 18)*. Baltimore, MD: USENIX Association, Aug. 2018, pp. 991–1008. ISBN: 978-1-939133-04-5. URL: <https://www.usenix.org/conference/usenixsecurity18/presentation/bulck>.
- [110] Runjie Zhang, Mircea R Stan, and Kevin Skadron. “HotSpot 6.0: Validation, acceleration and extension”. In: *University of Virginia, Tech. Rep* (2015).
- [111] Nathan Binkert et al. “The Gem5 Simulator”. In: *SIGARCH Comput. Archit. News* 39.2 (Aug. 2011), pp. 1–7. ISSN: 0163-5964. DOI: 10.1145/2024716.2024718. URL: <http://doi.acm.org/10.1145/2024716.2024718>.
- [112] Raymond Heald et al. “A third-generation SPARC V9 64-b microprocessor”. In: *IEEE JSSC* (2000).
- [113] A. Pellegrini and V. Bertacco. “Cobra: A comprehensive bundle-based reliable architecture”. In: *2013 International Conference on Embedded Computer Systems: Architectures, Modeling, and Simulation (SAMOS)*. July 2013, pp. 247–254.
- [114] C. H. Lin et al. “The effect of NBTI on 3D integrated circuits”. In: *2012 IEEE Electrical Design of Advanced Packaging and Systems Symposium (EDAPS)*. Dec. 2012, pp. 201–204. DOI: 10.1109/EDAPS.2012.6469435.
- [115] Jin Sun et al. “Workload Assignment Considering NBTI Degradation in Multicore Systems”. In: *J. Emerg. Technol. Comput. Syst.* 10.1 (Jan. 2014), 4:1–4:22. ISSN: 1550-4832. DOI: 10.1145/2539124. URL: <http://doi.acm.org/10.1145/2539124>.
- [116] F. Oboril and M. B. Tahoori. “Reducing wearout in embedded processors using proactive fine-grain dynamic runtime adaptation”. In: *2012 17th IEEE European Test Symposium (ETS)*. May 2012, pp. 1–6. DOI: 10.1109/ETS.2012.6233012.
- [117] K. W. Lee et al. “3D hetero-integration technology with backside TSV and reliability challenges”. In: *2013 IEEE SOI-3D-Subthreshold Microelectronics Technology*

*Unified Conference (S3S)*. Oct. 2013, pp. 1–2. DOI: 10.1109/S3S.2013.6716516.

- [118] S. V. Kumar, C. H. Kim, and S. S. Sapatnekar. “Adaptive Techniques for Overcoming Performance Degradation Due to Aging in CMOS Circuits”. In: *IEEE Transactions on VLSI Systems* (Apr. 2011).
- [119] Aaron D Franklin et al. “Variability in carbon nanotube transistors: Improving device-to-device consistency”. In: *ACS nano* 6.2 (2012), pp. 1109–1115.
- [120] Jie Zhang et al. “Characterization and design of logic circuits in the presence of carbon nanotube density variations”. In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 30.8 (2011), pp. 1103–1113.
- [121] Stanley E Schuster. “Multiple word/bit line redundancy for semiconductor memories”. In: *IEEE Journal of Solid-State Circuits* 13.5 (1978), pp. 698–703.
- [122] J. Zhang et al. “Overcoming CNT variations through co-optimized technology and circuit design”. In: *2011 International Electron Devices Meeting*. Dec. 2011, pp. 4.6.1–4.6.4.
- [123] Chen Wang et al. “Variation-Aware Global Placement for Improving Timing-Yield of Carbon-Nanotube Field Effect Transistor Circuit”. In: *ACM Trans. Des. Autom. Electron. Syst.* 23.4 (June 2018), 44:1–44:27. ISSN: 1084-4309. DOI: 10.1145/3175500. URL: <http://doi.acm.org/10.1145/3175500>.
- [124] Z. Ahmed et al. “Modeling CNTFET Performance Variation Due to Spatial Distribution of CNTs”. In: *IEEE Transactions on Electron Devices* (Sept. 2016), pp. 3776–3781. ISSN: 0018-9383.
- [125] B. Ghavami et al. “CNT-count Failure Characteristics of Carbon Nanotube FETs under Process Variations”. In: *2011 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems*. Oct. 2011, pp. 86–92. DOI: 10.1109/DFT.2011.31.
- [126] D. Blaauw et al. “Statistical Timing Analysis: From Basic Principles to State of the Art”. In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 27.4 (Apr. 2008), pp. 589–607. ISSN: 0278-0070. DOI: 10.1109/TCAD.2007.907047.
- [127] Aseem Agarwal et al. “Path-based statistical timing analysis considering inter-and intra-die correlations”. In: *Proc. TAU*. 2002, pp. 16–21.
- [128] P. Avouris. “Molecular Electronics with Carbon Nanotubes”. In: *Acc. Chem. Res.* (2002). URL: [http://pubs3.acs.org/acs/journals/doilookup?in\\_doi=10.1021/ar010152e](http://pubs3.acs.org/acs/journals/doilookup?in_doi=10.1021/ar010152e).

- [129] *Online STANFORD Virtual Source - CNFET model*. <https://nano.stanford.edu/stanford-cnfet2-model>.
- [130] A. S. Leon et al. “A Power-Efficient High-Throughput 32-Thread SPARC Processor”. In: *JSSC* 42.1 (Jan. 2007), pp. 7–16. ISSN: 0018-9200. DOI: 10.1109/JSSC.2006.885049.